
ansible-role-config-light Documentation

Release 2.1.0

Vladimir Botka

Oct 12, 2021

Table of Contents

1	Quick start guide	3
2	User's guide	9
2.1	Introduction	10
2.2	Installation	10
2.3	Playbook	11
2.4	Debug	11
2.5	Tags	12
2.6	Variables	12
2.7	Best practice	33
3	Examples	37
3.1	FreeBSD Postfix	38
3.2	Armbian Simple SMTP	44
4	Annotated source code	47
4.1	Tasks	47
4.2	Handlers	75
4.3	Templates	75
5	Collections	87
5.1	vbotka.freebsd	87
6	Copyright	89
7	Legal Notice	91
8	Indices and tables	93

This role and the documentation is work in progress. Feel free to [share your feedback](#) and [report issues](#).
Contributions are welcome.

Quick start guide

For users who want to quickly try the role this guide provides an example of how to install and configure [Lighttpd](#) on single FreeBSD host. The procedure is generic and can be easily modified to install and configure other applications on other systems. See examples in the directory *contrib*. The control node of this example is Linux and the user is a member of the group *adm*.

- Install the role `vbotka.config_light` and the collections

```
shell> ansible-galaxy role install vbotka.config_light
shell> ansible-galaxy collection install ansible.posix
shell> ansible-galaxy collection install community.general
```

- Create the playbook `config-light.yml` for single host `srv.example.com` (2) and the role (10)

```
1 shell> cat config-light.yml
2 - hosts: srv.example.com
3   gather_facts: true
4   connection: ssh
5   remote_user: admin
6   become: yes
7   become_user: root
8   become_method: sudo
9   roles:
10  - vbotka.config_light
```

- Create files in `host_vars` with customized variables of the role (2) and with the variables of the application (3).

```
1 shell> ls -1 host_vars/srv.example.com/config-light-*
2 host_vars/srv.example.com/config-light-common.yml
3 host_vars/srv.example.com/config-light-lighttpd.yml
```

- Create files in *host_vars* with customized variables of the role. To speedup the execution let's set the control-flow variables (2-5) to *false* and disable some steps. Enable these steps selectively when needed. The configuration files of the role will be stored in the directory *conf-light* in the current directory of the playbook (10). Set the

ownership and permissions of the directories on the control node so that the user who is running the playbook will be able both read and write the files, and create the directories (7-9) (11-14).

```

1 shell> cat host_vars/srv.example.com/config-light-common.yml
2 cl_sanity: false
3 cl_setup: false
4 cl_install: false
5 cl_debug: false
6 cl_backup: true
7 cl_dird_owner: "root"
8 cl_dird_group: "adm"
9 cl_dird_dmode: "0770"
10 cl_dird: "{{ playbook_dir }}/conf-light"
11 cl_dira_owner: "root"
12 cl_dira_group: "adm"
13 cl_dira_dmode: "0770"
14 cl_dira_fmode: "0660"

```

- Create files in *host_vars* with the variables of the application. Start the server (2), run the server at boot (3), and configure two files (5,18).

```

1 shell> cat host_vars/srv.example.com/config-light-lighttpd.yml
2 cl_service_lighttpd_enable: true
3 cl_service_lighttpd_state: 'started'
4
5 # /usr/local/etc/lighttpd/lighttpd.conf
6 cl_lighttpd_server_port: '80'
7 cl_lighttpd_server_useipv6: 'disable'
8 cl_lighttpd_server_username: 'www'
9 cl_lighttpd_server_groupname: 'www'
10 cl_lighttpd_server_document_root: "/usr/local/www/lighttpd"
11 cl_lighttpd_lighttpdconf_dict:
12   - {key: 'server.port', value: "{{ cl_lighttpd_server_port }}" }
13   - {key: 'server.use-ipv6', value: "{{ cl_lighttpd_server_useipv6 }}" }
14   - {key: 'server.username', value: "{{ cl_lighttpd_server_username }}" }
15   - {key: 'server.groupname', value: "{{ cl_lighttpd_server_groupname }}" }
16   - {key: 'server.document-root', value: "{{ cl_lighttpd_server_document_root }}" }
17
18 # /etc/rc.conf
19 cl_lighttpd_rcconf_lighttpd_enable: 'YES'
20 cl_lighttpd_rcconf_dict:
21   - {key: 'lighttpd_enable', value: "{{ cl_lighttpd_rcconf_lighttpd_enable }}" }

```

- Create configuration files in the directory *conf-light*

```

1 shell> tree conf-light
2 conf-light/
3 └─ files.d
4     └─ lighttpd-lighttpdconf
5         └─ lighttpd-rcconf
6 └─ handlers.d
7     └─ lighttpd-freebsd
8 └─ packages.d
9     └─ lighttpd
10 └─ services.d
11     └─ lighttpd
12 └─ states.d
13     └─ lighttpd-server-document-root

```


conf-light/files.d

```

1 shell> cat conf-light/files.d/lighttpd-lighttpdconf
2 lighttpd-lighttpdconf:
3   path: '/usr/local/etc/lighttpd/lighttpd.conf'
4   create: true
5   owner: 'root'
6   group: 'wheel'
7   mode: '0644'
8   assignment: ' = '
9   dict: '{{ cl_lighttpd_lighttpdconf_dict }}'
10  handlers:
11    - 'reload lighttpd'

```

```

1 shell> cat conf-light/files.d/lighttpd-rconf
2 lighttpd_rconf:
3   path: '/etc/rc.conf'
4   create: true
5   owner: 'root'
6   group: 'wheel'
7   mode: '0644'
8   assignment: '='
9   dict: "{{ cl_lighttpd_rconf_dict }}"
10  handlers:
11    - 'reload lighttpd'

```

conf-light/handlers.d

```

1 shell> cat conf-light/handlers.d/lighttpd-freebsd
2 lighttpd_freebsd:
3   template: handlers-auto2.yml.j2
4   handlers:
5
6   - handler: 'enable and start lighttpd'
7     module: service
8     params:
9       - 'name: lighttpd'
10      - 'state: started'
11      - 'enabled: true'
12
13  - handler: 'disable and stop lighttpd'
14    module: service
15    params:
16      - 'name: lighttpd'
17      - 'state: stopped'
18      - 'enabled: false'
19
20  - handler: 'reload lighttpd'
21    module: service
22    params:
23      - 'name: lighttpd'
24      - 'state: reloaded'
25    conditions:
26      - '- cl_service_lighttpd_enable|bool'
27
28  - handler: 'restart lighttpd'
29    module: service
30    params:

```

(continues on next page)

(continued from previous page)

```

31     - 'name: lighttpd'
32     - 'state: restarted'
33     conditions:
34     - '- cl_service_lighttpd_enable|bool'
35
36     - handler: 'lighttpd check'
37     module: command
38     params:
39     - 'cmd: /usr/local/sbin/lighttpd -t'

```

conf-light/packages.d

```

1 shell> cat conf-light/packages.d/lighttpd
2 lighttpd:
3   name:
4   - 'www/lighttpd'

```

conf-light/services.d

```

1 shell> cat conf-light/services.d/lighttpd
2 lighttpd:
3   name: 'lighttpd'
4   state: '{{ cl_service_lighttpd_state }}'
5   enabled: '{{ cl_service_lighttpd_enable }}'

```

conf-light/states.d

```

1 shell> cat conf-light/states.d/lighttpd-server-document-root
2 lighttpd_server_document_root:
3   state: directory
4   path: '{{ cl_lighttpd_server_document_root }}'
5   owner: '{{ cl_lighttpd_server_username }}'
6   group: '{{ cl_lighttpd_server_groupname }}'
7   mode: '0750'

```

- Enable setup and create variables

```
shell> ansible-playbook config-light.yml -t cl_vars -e 'cl_setup=true'
```

This command will assemble the configuration data and create handlers on the control node. Take a look at directory `conf-light/assemble/` what files were created. Also take a look at the directory `roles/vbotka.config_light/handlers` what handlers were created.

- Enable and test sanity

```
shell> ansible-playbook config-light.yml -t cl_sanity -e 'cl_sanity=true'
```

- Display variables

```
shell> ansible-playbook config-light.yml -t cl_debug -e 'cl_debug=true'
```

- Install packages

```
shell> ansible-playbook config-light.yml -t cl_packages -e 'cl_install=true'
```

- Set states of the files

```
shell> ansible-playbook config-light.yml -t cl_states
```

- Create and modify files

```
shell> ansible-playbook config-light.yml -t cl_files
```

- Configure services

```
shell> ansible-playbook config-light.yml -t cl_services
```

- The role and the configuration data in the examples are idempotent. Once the application is installed and configured there should be no changes reported by *ansible-playbook* when running the playbook repeatedly. Disable setup, sanity, debug, and install to speedup the playbook

```
1 shell> ansible-playbook config-light.yml
2
3 [...]
4
5 PLAY RECAP
6 ↪*****
  srv.example.com: ok=21 changed=0 unreachable=0 failed=0 skipped=35 rescued=0
  ↪ignored=0
```

- Create file `/usr/local/www/lighttpd/index.html`

```
1 shell> ll /usr/local/www/lighttpd/index.html
2 -rw-r--r-- 1 www www 51 Apr 12 18:58 /usr/local/www/lighttpd/index.html
3 shell> cat /usr/local/www/lighttpd/index.html
4 <html><body><h1>Lighttpd works!</h1></body></html>
```

- Open the page in a browser `http://srv.example.com/`. The content should be

```
Lighttpd works!
```


Table of Contents

- *User's guide*
 - *Introduction*
 - *Installation*
 - *Playbook*
 - *Debug*
 - *Tags*
 - *Variables*
 - *Best practice*
 - * *Check syntax of setup*
 - * *Setup*
 - * *Validation*
 - * *Check syntax*
 - * *Manage packages*
 - * *Manage states of files*
 - * *Manage configuration files*
 - * *Manage services*
 - * *Idempotency*

2.1 Introduction

- Ansible role: `config_light`
- Supported systems: FreeBSD, Ubuntu
- Requirements: `ansible.posix` , `community.general`

The role installs packages, creates and configures files, services, and handlers. This provides a simple, but flexible framework to apply basic Ansible modules. A substantial part of the control-flow will be determined by the structure of the data. Some attributes of the dictionaries trigger Ansible modules to modify configuration files, configure services and create handlers.

The role can be used with any supported OS to install and configure arbitrary applications. The role is tested with supported releases of FreeBSD and Ubuntu. It can be expected that other BSD and Linux distributions, that support the Ansible modules mentioned below, should work with minimal changes. Red Hat and Debian *ansible_os_family* should work out of the box.

Ansible modules `package`, `apt`, `yum`, and `snap` are used to install Linux packages. In FreeBSD, modules `pkgng` and `portinstall` are used to install FreeBSD packages and ports.

Ansible modules `file`, `template`, `copy`, `replace`, `patch`, `lineinfile`, `blockinfile`, and `ini_file` are used to configure files. The module `mount` is used to mount and unmount paths, and to configure *fstab*. Module `service` is used to manage both Linux and FreeBSD services.

The directory `contrib` comprises examples on how to install and configure various applications and how to create the handlers and templates. Some of them are commented *Examples*.

The user of this role is expected to master at least the following Ansible topics

- [Basic Concepts](#)
- [Roles](#)
- [Working With Playbooks](#)

Feel free to [share your feedback](#) and [report issues](#). The contributions to the [project](#) are welcome.

2.2 Installation

The most convenient way how to install an Ansible role is to use Ansible Galaxy CLI `ansible-galaxy`. The utility comes with the standard Ansible package and provides the user with a simple interface to the Ansible Galaxy's services. For example, take a look at the current status of the role

```
shell> ansible-galaxy role info vbotka.config_light
```

and install it

```
shell> ansible-galaxy role install vbotka.config_light
```

Install the collections `community.general` and `ansible.posix`

```
shell> ansible-galaxy collection install ansible.posix
shell> ansible-galaxy collection install community.general
```

Optionally install package `ansible-lint` if you want to enable the validation of created handlers and assembled data.

See also:

- To install specific versions from various sources see [Installing content](#)
- Take a look at other roles `shell> ansible-galaxy search --author=vbotka`

2.3 Playbook

Below is a simple playbook that calls this role (10) at a single host `srv.example.com` (2)

```
1 shell> cat config-light.yml
2 - hosts: srv.example.com
3   gather_facts: true
4   connection: ssh
5   remote_user: admin
6   become: yes
7   become_user: root
8   become_method: sudo
9   roles:
10  - vbotka.config_light
```

Note: `gather_facts: true` (3) must be set to gather facts needed to evaluate OS-specific options of the role. For example to install packages the variable `ansible_os_family` is needed to select the appropriate Ansible module.

See also:

- For details see [Connection Plugins](#) (4-5)
- and [Understanding Privilege Escalation](#) (6-8)

2.4 Debug

To see additional debug information enable debug output in the configuration

```
cl_debug: true
```

, or set the extra variable in the command:

```
shell> ansible-playbook config-light.yml -e 'cl_debug=true'
```

Note: The debug output of this role is optimized for the `yaml` callback plugin. Set this plugin, for example, in the environment `shell> export ANSIBLE_STDOUT_CALLBACK=yaml`.

See also:

- [Playbook Debugger](#)
- [Debugging modules](#)
- [Python Debugging With Pdb](#)

2.5 Tags

The tags provide a very useful tool to run selected tasks of the role. To see what tags are available list the tags of the role with the command

```
1 shell> ansible-playbook config-light.yml --list-tags
2
3 playbook: config-light.yml
4
5 play #1 (srv.example.com): srv.example.com TAGS: []
6 TASK TAGS: [always, cl_debug, cl_files, cl_packages, cl_sanity,
7 cl_services, cl_setup, cl_states, cl_vars]
```

For example, display the list of the variables and their values with the tag `cl_debug` (when debug is enabled `cl_debug: true`). With this tag specified `-t cl_debug` all imported tasks before the task `debug.yml` will also run because of the tag `always` when sanity testing is enabled `cl_sanity: true` (default) and setup is enabled `cl_setup: true` (default).

```
shell> ansible-playbook config-light.yml -t cl_debug
```

See what packages will be installed

```
shell> ansible-playbook config-light.yml -t cl_packages --check
```

Install packages and exit the play

```
shell> ansible-playbook config-light.yml -t cl_packages
```

See also:

- See [Best practice](#) on how to use tags efficiently
- See [main.yml](#) annotated source code
- See [main.yml](#) at GitHub

2.6 Variables

The *Default variables* control the options of the role. Most important are the variables that control the collection of the configuration data. For example, in each project, customize the directory `cl_dird` where the files with the configuration data are stored.

See the particular sections below on how to configure the creation of handlers, installation of the packages or ports, and management of files and services. Most options are available in the section *Files*. See the particular subsections on how to create the configuration data for the Ansible modules that serve the options. Review hints in the *Examples*.

See also:

- See [vars.yml](#) at GitHub
- See [vars.yml](#) annotated source code
- [Ansible variable precedence: Where should I put a variable?](#)

Note: The names of the dictionaries in the configuration files `cl_dird/*.d/*` are not used by the role and can be any arbitrary string that is a valid name of an Ansible variable. The name must be unique in the particular option

(directory files.d, packages.d, ...).

2.6.1 Default variables

Default variables are stored in the directory `defaults`.

Most of the variables are self-explaining. There are five very important variables `cl_handlers`, `cl_packages`, `cl_states`, `cl_services`, and `cl_files` (13-17). These dictionaries, which comprise the configuration data of handlers, packages, services, and files, will be explained in details. By default, these dictionaries are empty.

Best practice is to provide the data either in `host_vars` and `group_vars` or as a files in the directories `cl_handlersd_dir`, `cl_packagesd_dir`, `cl_statesd_dir`, `cl_servicesd_dir`, and `cl_filesd_dir` (24-28). Both methods can be applied at the same time. The variables will be assembled and combined by the tasks `vars_handlers.yml`, `vars_packages.yml`, `vars_states.yml`, `vars_services.yml`, and `vars_files.yml`. The assembled dictionaries, customized for each host in the play, will be stored in the host-specific files `cl_packagesd`, `cl_statesd`, `cl_servicesd`, and `cl_filesd` (40-43). The variable `cl_handlers` is not host-specific, because the handlers will be created at the controller (localhost) only. Assembled dictionary `cl_handlers` will be stored in the file `cl_handlersd` (39). Take a look at the directory `cl_dira` (38) to see assembled data.

By default, the base of the directories is `role_path` (23). The user is expected to put the configuration data to a more suitable directory, e.g., to `playbook_dir` directory.

[defaults/main.yml]

```

1  ---
2  # defaults for config_light
3
4  cl_sanity: true           # Import tasks/sanity.yml
5  cl_setup: true           # Import tasks/setup.yml
6  cl_install: true         # Install packages or ports
7  cl_debug: false          # Print debug output
8  cl_backup: false         # Backup files
9  cl_copyfile_delete: false # Delete dest then copy samples and defaults
10 cl_template_delete: false # Delete dest then create from templates
11
12 # Combine assembled data with these variables
13 cl_handlers: {}
14 cl_packages: {}
15 cl_services: {}
16 cl_files: {}
17 cl_states: {}
18
19 # Assemble data from these directories
20 # cl_dird_owner: root      # no default
21 # cl_dird_group: adm      # no default
22 cl_dird_dmode: "0775"    # default very permissive, restrict if necessary
23 cl_dird: "{{ role_path }}/files"
24 cl_handlersd_dir: "{{ cl_dird }}/handlers.d"
25 cl_packagesd_dir: "{{ cl_dird }}/packages.d"
26 cl_servicesd_dir: "{{ cl_dird }}/services.d"
27 cl_filesd_dir: "{{ cl_dird }}/files.d"
28 cl_statesd_dir: "{{ cl_dird }}/states.d"
29
30 # Lint
31 cl_yamllint_missing_fatal: true

```

(continues on next page)

(continued from previous page)

```

32 cl_yamllint: yamlLint
33 cl_yamllint_rules:
34     extends: default
35     rules:
36         line-length:
37             level: warning
38 cl_assemble_validate: "{{ cl_yamllint }}" -d '{{ cl_yamllint_rules|to_json }}' %s"
39 cl_handlers_validate: "{{ cl_yamllint }}" -d '{{ cl_yamllint_rules|to_json }}' %s"
40
41 # Assemble inventory_hostname data into these files
42 # cl_dira_owner: root           # no default
43 # cl_dira_group: adm           # no default
44 cl_dira_dmode: "0775"         # default very permissive, restrict if necessary
45 cl_dira_fmode: "0664"         # default very permissive, restrict if necessary
46 cl_dira: "{{ cl_dird }}/assemble"
47 cl_handlersd: "{{ cl_dira }}/handlersd" # localhost; not inventory_hostname specific
48 cl_packagesd: "{{ cl_dira }}/packagesd.{{ inventory_hostname }}"
49 cl_servicesd: "{{ cl_dira }}/servicesd.{{ inventory_hostname }}"
50 cl_filesd: "{{ cl_dira }}/filesd.{{ inventory_hostname }}"
51 cl_statesd: "{{ cl_dira }}/statesd.{{ inventory_hostname }}"
52 cl_assemble_regexp: '^(.*)[^\~]$' # Any string but terminated by ~
53
54 # Role handlers directory
55 # cl_handlers_dir_owner: admin # no default
56 # cl_handlers_dir_group: admin # no default
57 # cl_handlers_dir_dmode: "0775" # no default
58 # cl_handlers_main_mode: "0644" # no default
59 cl_handlers_clean_all: false
60 cl_handlers_create: true
61 cl_handlers_delete: false
62 cl_handlers_dir_become: false
63
64 # Snap
65 cl_snap_paths:
66     - /usr/local/sbin
67     - /usr/local/bin
68     - /usr/sbin
69     - /usr/bin
70     - /sbin
71     - /bin
72     - /snap/bin
73 cl_snap_patterns:
74     - snap
75
76 # States
77 cl_states_unmount: [absent, unmounted]
78 cl_states_mount: [present, mounted, remounted]
79 cl_states_file: [absent, directory, file, hard, link, touch]
80
81 # Files
82 cl_files_order: [copy, template, markers, create-backup, patch, lineinfile,
83     ↪blockinfile, inifile, ucl, delete-backup]
84
85 # OS common
86 install_retries: 10
87 install_delay: 5

```

(continues on next page)

(continued from previous page)

```

88 # FreeBSD
89 freebsd_install_method: packages
90 # freebsd_install_method: ports
91 freebsd_use_packages: true
92 cl_services_freebsd_rcconf_auto: false
93
94 # EOF
95 ...

```

Warning: Defaults of the variables *cl_dird_dmode* (22), *cl_dira_dmode* (36) and *cl_dira_fmode* (37) are very permissive. These are the permissions to access the configuration data and the assembled dictionaries. Restrict the permissions if these dictionaries might comprise classified data.

<TODO: complete description of all default variables>

2.6.2 Handlers

- *Synopsis*
- *Parameters*
- *Example*
- *See Also*
- *Notes*

Synopsis

The dictionary *cl_handlers* comprises data to create handlers. The structure of the dictionary depends on the template that is used to create the files with the handlers. For example, the structure below can be used together with the template *handlers-auto1.yml.j2*.

Parameters

Parameter	Type	Comments	
<code>template</code>	string required	Template filename	
<code>handlers</code>	list required	List of handlers dictionaries	
	<code>handler</code>	string required	Name of the handler
	<code>module</code>	string required	Ansible module in handler
	<code>params</code>	list required	Ansible module parameters
	<code>conditions</code>	list	List of conditions

Example

FreeBSD handlers for Postfix

[contrib/postfix/conf-light/handlers.d/postfix-freebsd.yml]

```

1 postfix_freebsd:
2   template: handlers-autom1.yml.j2
3   handlers:
4
5     - handler: 'enable and start postfix'
6       module: service
7       params:
8         - 'name: postfix'
9         - 'state: started'
10        - 'enabled: true'
11
12    - handler: 'disable and stop postfix'
13      module: service
14      params:
15        - 'name: postfix'
16        - 'state: stopped'
17        - 'enabled: false'
18
19    - handler: 'reload postfix'
20      module: service
21      params:
22        - 'name: postfix'
23        - 'state: reloaded'
24      conditions:
25        - '- cl_service_postfix_enable|bool'
26
27    - handler: 'restart postfix'
28      module: service
29      params:
30        - 'name: postfix'
31        - 'state: restarted'
32      conditions:
33        - '- cl_service_postfix_enable|bool'
34
35    - handler: 'postfix check'
36      module: command
37      params:
38        - 'cmd: /usr/local/sbin/postfix check'
39
40    - handler: 'newaliases'
41      module: command
42      params:
43        - 'cmd: /usr/bin/newaliases'
44
45    # - handler: 'postmap smtp sasl passwords'
46    #   module: command
47    #   params:
48    #     - 'cmd: /usr/local/sbin/postmap {{ postfix_main_cf_smtp_sasl_password_maps }}'
49    ↪
50
51    # - handler: 'postmap virtual aliases'
52    #   module: command
53    #   params:
54    #     - 'cmd: /usr/local/sbin/postmap {{ postfix_virtual }}'
    
```

See Also

See also:

- For details see the template `handlers-auto1.yml.j2`
- See `vars-handlers.yml` at GitHub how the variable `cl_handlers` is combined with the content of the directory `cl_handlersd_dir`
- See `setup.yml` at GitHub how the handlers are created

Annotated source code

- `vars-handlers.yml`
- `setup.yml`

Notes

Note: The template `handlers-auto1.yml.j2` is available in the role's directory `templates`. The user is expected to create new templates when needed. Feel free to change the structure of the data and to create new templates that might fit the purpose better. Feel free to contribute new templates and configuration examples to the [project](#).

2.6.3 Packages

- *Synopsis*
 - *FreeBSD*
 - *snap*
- *Parameters*
- *Examples*
- *See Also*

Synopsis

The dictionary `cl_packages` comprises managed packages (Linux or BSD) or BSD ports.

FreeBSD

By default packages will be installed. If you want to install ports set

```
freebsd_install_method: ports
```

snap

By default snap packages won't be installed or uninstalled if `snap` binary can't be found in `cl_snap_paths`. If you want the role to fail when `snap` is missing set

```
cl_snap_missing_fatal: true
```

The variables `cl_snap_missing_fatal`, `cl_snap_paths`, `cl_snap_patterns` are declared in `defaults/main.yml`.

Parameters

Parameter	Type	Comments
<code>name</code>	list required	List of packages or BSD ports
<code>module</code>	string	Ansible module to manage Linux packages (default=package) choices: package, yum, apt, snap
<code>state</code>	string	State of packages or BSD ports (default=present)
...	...	<TBD: see tasks/packages.yml>

Examples

- FreeBSD install Postfix package or port

[contrib/postfix/conf-light/packages.d/postfix.yml]

```
1 postfix:
2   name:
3     - 'mail/postfix'
```

- Armbian package for Simple SMTP

[contrib/ssmtp/conf-light/packages.d/ssmtp.yml]

```
1 ssmtp:
2   name:
3     - 'ssmtp'
```

- Ubuntu delete snap packages

[contrib/ubuntu-snap-disable/conf-light/packages.d/snap-deinstall.yml]

```
1 snap_deinstall_list1:
2   module: snap
3   state: absent
4   name:
5     - chromium
6     - core
7     - core18
8     - gnome-3-28-1804
9     - gtk-common-themes
10    - simplenote
11    - tusk
12    # - snapd
13    #
14    # msg: Oops! Snap installation failed while executing 'sh -c "/usr/bin/snap remove_
15    ↪ chromium core
16    # core18 gnome-3-28-1804 gtk-common-themes simplenote snapd tusk"', please examine_
17    ↪ logs and error
```

(continues on next page)

(continued from previous page)

```

16 # output for more details.
17 # rc: 1
18 # stderr: |-
19 #   error: cannot remove "chromium", "core", "core18", "gnome-3-28-1804",
20 #       "gtk-common-themes", "simplenote", "snapd", "tusk": snap "snapd" is not
21 #       removable: remove all other snaps first

```

- Ubuntu purge snapd package

[contrib/ubuntu-snap-disable/conf-light/packages.d/snapd.yml]

```

1 snap_deinstall_list2:
2   module: apt
3   state: absent
4   purge: true
5   name:
6     - snapd

```

See Also

See also:

- See [vars-packages.yml](#) at GitHub how the variable `cl_packages` is combined with the content of the directory `cl_packagesd_dir`
- See [packages.yml](#) at GitHub how the packages or BSD ports are installed

Annotated source code

- [vars-packages.yml](#)
- [packages.yml](#)

2.6.4 States

- [Synopsis](#)
- [Parameters](#)
- [Examples](#)
- [See Also](#)

Synopsis

The dictionary `cl_states` comprises the states of the managed files. If mounted, the `path` is unmounted when `state` is in the list `cl_states_unmount` (default=absent)

```
cl_states_unmount: [absent, unmounted]
```

Then, the module `file` is applied if `state` is in the list `cl_states_file` (default=file)

```
cl_states_file: [absent, directory, file, hard, link, touch]
```

In the end, the *path* is mounted if *state* is in the list `cl_states_mount` (default=absent)

```
cl_states_mount: [present, mounted, remounted]
```

The variables `cl_states_unmount`, `cl_states_mount`, `cl_states_file` are declared in `defaults/main.yml`. Details of the parameters are described in the modules `mount` and `file`.

Parameters

Parameter	Type	Comments
<code>path</code>	string required	Path to file
<code>state</code>	string	State of the file
<code>owner</code>	string	Owner of the file
<code>group</code>	string	Group of the file
<code>mode</code>	string	Mode of the file
...	...	<TBD: see tasks/states.yml>

Examples

- Ownership and permissions of the document root for Lighttpd

[contrib/lighttpd/conf-light/states.d/lighttpd-server-document-root.yml]

```
1 lighttpd_server_document_root:
2   state: directory
3   path: '{{ cl_lighttpd_server_document_root }}'
4   owner: '{{ cl_lighttpd_server_username }}'
5   group: '{{ cl_lighttpd_server_groupname }}'
6   mode: '0640'
```

- Delete snap directories

[contrib/ubuntu-snap-disable/conf-light/states.d/snap.yml]

```
1 snap_root:
2   path: /snap
3   state: absent
4 snap_var:
5   path: /var/snap
6   state: absent
7 snap_lib_var:
8   path: /var/lib/snapd
9   state: absent
```

See Also

See also:

- See [vars-states.yml](#) at GitHub how the dictionary `cl_states` is combined from the content of the directory `cl_statesd_dir`
- See [states.yml](#) at GitHub how the file's states are set
- See `shell> ansible-doc -t module mount`

- See `shell> ansible-doc -t module file`

Annotated source code

- `vars-states.yml`
- `states.yml`

2.6.5 Services

- *Synopsis*
- *Parameters*
- *Example*
- *See Also*

Synopsis

The dictionary `cl_services` comprises managed services.

Parameters

Parameter	Type	Comments
<code>name</code>	string required	Name of the service
<code>state</code>	string	State of the service default: started
<code>enabled</code>	boolean	Start on boot default: true

Example

FreeBSD services for Postfix and Sendmail

[contrib/postfix/conf-light/service.d/postfix.yml]

```

1 postfix:
2   name: 'postfix'
3   state: '{{ cl_service_postfix_state }}'
4   enabled: '{{ cl_service_postfix_enable }}'
```

[contrib/postfix/conf-light/service.d/sendmail.yml]

```

1 sendmail:
2   name: 'sendmail'
3   state: '{{ cl_service_sendmail_state }}'
4   enabled: '{{ cl_service_sendmail_enable }}'
```

See Also

See also:

- See [vars-services.yml](#) at GitHub how the variable `cl_services` is combined with the content of the directory `cl_servicesd_dir`
- See [services.yml](#) at GitHub how the services are configured

Annotated source code

- [vars-services.yml](#)
- [services.yml](#)

2.6.6 Files

- [Synopsis](#)
- [Order of the options](#)
- [See Also](#)
- [Options](#)

Synopsis

The variable `cl_files` is a dictionary of the files that shall be managed by this role. It's optional which Ansible module will be used to manage a file. More options can be applied at the same file. For example, it is possible to create a file by the Ansible module `template` and modify it by the module `lineinfile` later. Several options, listed in the default order, are available

1. `copy`: If the attribute `copyfile` is defined in the dictionary
2. `template`: If the attribute `template` is defined in the dictionary
3. `create blockinfile markers`: If the attribute `markers` is defined in the dictionary
4. `patch`: If the attribute `patch` is defined in the dictionary
5. `lineinfile`: If the attribute `dict` or `lines` is defined in the dictionary
6. `blockinfile`: If the attribute `blocks` is defined in the dictionary
7. `ini_file`: If the attribute `ini` is defined in the dictionary

Order of the options

The variable `cl_files_order` controls the order of the execution. Multiple options, when present in the dictionary of a file definition, will be applied in this order. In addition to the options, listed above, there are `create-backup` and `delete-backup` tasks to backup files that was changed if enabled by `cl_backup` (default: `false`). By default, the backup files are created after `copy`, `template`, and `markers`. Fit the order of the execution to your needs.

See Also

See also:

- See [vars-files.yml](#) at GitHub how the variable `cl_files` is combined with the content of the directory `cl_filesd_dir`
- See [files.yml](#) at GitHub how the files are created and modified

- See `files-create-backup.yml` at GitHub how the backups are created (when enabled by `cl_backup`)
- See `files-delete-backup.yml` at GitHub how the backup files are deleted when the files haven't been modified

Annotated source code

- `vars-files.yml`
- `files.yml`
- `files-create-backup.yml`
- `files-delete-backup.yml`

Options

copy

Copy files.

Parameters for copyfile

Parameter	Type	Comments
<code>path</code>	string required	Path to file
<code>copyfile</code>	dict required	Copyfile parameters (see <code>tasks/files-copy.yml</code>)
	<code>path</code>	string required
	<code>remote_src</code>	string
	<code>force</code>	boolean
	<code>...</code>	<code>...</code>
<code>owner</code>	string	Owner of the file
<code>group</code>	string	Group of the file
<code>mode</code>	string	Mode of the file
<code>attributes</code>	string	Attributes of the file
<code>validate</code>	string	Command to validate file
<code>handlers</code>	list	List of handlers

Example of copyfile

Create the description of the file (2) and declare the variable for the dictionary (7)

[`contrib/httptpd_nagios/conf-light/files.d/httptpd-modulesconf.yml`]

```

1 httptpd-modulesconf:
2   path: '/usr/local/etc/httptpd/modules.conf'
3   create: true
4   owner: 'root'
5   group: 'wheel'
6   mode: '0644'
7   copyfile: '{{ cl_httptpd_modulesconf_copy }}'
8   markers: '{{ cl_httptpd_modulesconf_markers }}'
9   lines: '{{ cl_httptpd_modulesconf_lines }}'
10  blocks: '{{ cl_httptpd_modulesconf_blocks }}'
11  handlers:

```

(continues on next page)

(continued from previous page)

```
12 - 'configtest lighttpd'
13 - 'reload lighttpd'
```

Create the dictionary `cl_lighttpd_modulesconf_copy` (69)

[contrib/lighttpd_nagios/cl-lighttpd.yml]

```
68 # /usr/local/etc/lighttpd/modules.conf
69 cl_lighttpd_modulesconf_copy:
70   path: '/usr/local/etc/lighttpd/modules.conf.sample'
71   remote_src: true
72   force: false
```

Then, the command

```
shell> ansible-playbook config-light.yml -t cl_files_copy
```

will copy sample file `modules.conf.sample` to `modules.conf` if the destination does not exist.

See Also

See also:

- See [files-copy.yml](#) at GitHub
- See [files-copy.yml](#) annotated source code

template

Create files from templates.

Parameters for template

Parameter	Type	Comments
<code>path</code>	string required	Path to file
<code>template</code>	dict required	Template parameters (see files-templates.yml)
	<code>path</code>	string required
	<code>force</code>	boolean
	<code>...</code>	<code>...</code>
<code>owner</code>	string	Owner of the file
<code>group</code>	string	Group of the file
<code>mode</code>	string	Mode of the file
<code>attributes</code>	string	Attributes of the file
<code>validate</code>	string	Command to validate file
<code>handlers</code>	list	List of handlers

Example of template

File `/etc/mail/mailer.conf` for postfix

[contrib/postfix/conf-light/files.d/mailer-conf.yml]

```

1 mailerconf:
2   path: '/etc/mail/mailer.conf'
3   template:
4     path: 'mailer.conf.j2'
5     force: true
6   owner: 'root'
7   group: 'wheel'
8   mode: '0644'

```

See Also

See also:

- See [files-template.yml](#) how the files are modified or created by the Ansible module `template`
- See [files-template.yml](#) annotated source code

Notes

Note: There are couple of templates ready to be used in the directory `templates`. The user is expected to create new templates when needed. Feel free to contribute new templates to the [project](#).

blockinfile markers

Create markers for Ansible module `blockinfile`. Mark existing blocks that you want to configure.

Parameters for blockinfile markers

Parameter	Type	Comments
<code>path</code>	string required	Path to file
<code>markers</code>	list required	List of dictionaries (see <code>fn/mark-block.yml</code>)
	<code>regex1</code>	string required
	<code>replace1</code>	string required
	<code>regex2</code>	string required
	<code>replace2</code>	string required

Example of blockinfile markers

For example, in file `/usr/local/etc/lighttpd/modules.conf`, create `blockinfile` markers in the following block

```

1 ##
2
3 server.modules = (
4   "mod_access",
5   # "mod_alias",

```

(continues on next page)

(continued from previous page)

```

6 # "mod_auth",
7 # "mod_authn_file",
8 # "mod_evasive",
9 # "mod_setenv",
10 # "mod_usertrack",
11 # "mod_redirect",
12 # "mod_rewrite",
13 )
14
15 ##

```

Create the description of the file (2) and declare the variable for the list of the markers (8)

[contrib/lighttpd_nagios/conf-light/files.d/lighttpd-modulesconf.yml]

```

1 lighttpd-modulesconf:
2   path: '/usr/local/etc/lighttpd/modules.conf'
3   create: true
4   owner: 'root'
5   group: 'wheel'
6   mode: '0644'
7   copyfile: '{{ cl_lighttpd_modulesconf_copy }}'
8   markers: '{{ cl_lighttpd_modulesconf_markers }}'
9   lines: '{{ cl_lighttpd_modulesconf_lines }}'
10  blocks: '{{ cl_lighttpd_modulesconf_blocks }}'
11  handlers:
12    - 'configtest lighttpd'
13    - 'reload lighttpd'

```

Create the list of the dictionaries cl_lighttpd_modulesconf_markers (73)

[contrib/lighttpd_nagios/cl-lighttpd.yml]

```

68 # /usr/local/etc/lighttpd/modules.conf
69 cl_lighttpd_modulesconf_copy:
70   path: '/usr/local/etc/lighttpd/modules.conf.sample'
71   remote_src: true
72   force: false
73 cl_lighttpd_modulesconf_markers:
74 - marker: 'server.modules'
75   regex1: 'server.modules\s*=\s*\('
76   replacel: 'server.modules = ('
77   regex2: '\)'
78   replace2: ')'

```

Then, the command

```
shell> ansible-playbook config-light.yml -t cl_files_copy,cl_files_markers
```

will copy sample file modules.conf.sample to modules.conf and will create blockinfile markers

```

1 ##
2
3 # BEGIN ANSIBLE MANAGED BLOCK server.modules
4 server.modules = (
5     "mod_access",
6     # "mod_alias",

```

(continues on next page)

(continued from previous page)

```

7 # "mod_auth",
8 # "mod_authn_file",
9 # "mod_evasive",
10 # "mod_setenv",
11 # "mod_usertrack",
12 # "mod_redirect",
13 # "mod_rewrite",
14
15 )
16 # END ANSIBLE MANAGED BLOCK server.modules
17
18 ##

```

See Also

See also:

- See [files-markers.yml](#) at GitHub how the markers are created
- See [fn/mark-block.yml](#) at GitHub how the markers are created

Annotated source code

- [files-markers.yml](#)
- [mark-block.yml](#)

patch

Patch files.

Parameters for patch

Parameter	Type	Comments
path	string required	Path to file to be patched
patch	dict required	Parameters of patch module (see files-patch.yml)
	src	string required
	basedir	path

handlers	list	List of handlers

Example of patch

File `/etc/network.subr` for `/etc/rc.d/wpa_cli`

[[contrib/freebsd-custom-image-wpacli/conf-light/files.d/network_subr.yml](#)]

```

1 network_subr:
2   path: '{{ bsd_cimage_mount_path }}/etc/network.subr'
3   patch:

```

(continues on next page)

(continued from previous page)

```
4 src: '{{ playbook_dir }}/files/network.subr.patch'  
5 basedir: '{{ bsd_cimage_mount_path }}/etc'
```

See Also

See also:

- See [files-patch.yml](#) how the files are modified by the Ansible module `patch`
- See [files-patch.yml](#) annotated source code
- See details about the example in the role `vbotka.freebsd_wpa_cli`

lineinfile

Create or configure lines in a file.

Parameters for lineinfile

Parameter	Type	Comments	
path	string required	Path to file	
lines	list required	Lineinfile params. Either dict or lines is required. (see files-lineinfile.yml)	
	regexp	string required	Regular expression
	list	string required	Line
	backrefs		
	state		
	firstmatch		
	insertafter		
	insertbefore		
...	...	<TBD>	
assignment	string	Assignment of key and value in dict	
dict	list required	Lineinfile params. Either dict or lines is required. (see files-lineinfile.yml)	
	key	string required	Key value for regexp
	value	string required	Value for line
	firstmatch		
	insertafter		
	insertbefore		
...	...	<TBD>	
owner	string	Owner of the file	
group	string	Group of the file	
mode	string	Mode of the file	
attributes	string	Attributes of the file	
other	string	Attributes of module file	
create	boolean	Create file	
validate	string	Command to validate file	
handlers	list	List of handlers	

Example of lineinfile with lines

File `/usr/local/etc/lighttpd/lighttpd.conf` for `lighttpd`

[`contrib/lighttpd/conf-light/files.d/lighttpd-lighttpdconf-lines.yml`]

```

1 lighttpd-lighttpdconf:
2   path: '/usr/local/etc/lighttpd/lighttpd.conf'
3   create: true
4   owner: 'root'
5   group: 'wheel'
6   mode: '0644'
7   handlers:

```

(continues on next page)

(continued from previous page)

```

8   - 'reload lighttpd'
9   lines:
10  - regexp: '^\\s*server.port\\s*=\\s*(.*)$'
11    line: 'server.port = "80"'
12  - regexp: '^\\s*server.use-ipv6\\s*=\\s*(.*)$'
13    line: 'server.use-ipv6 = "disable"'
14  - regexp: '^\\s*server.username\\s*=\\s*(.*)$'
15    line: 'server.username = "www"'
16  - regexp: '^\\s*server.groupname\\s*=\\s*(.*)$'
17    line: 'server.groupname = "www"'
18  - regexp: '^\\s*server.document-root\\s*=\\s*(.*)$'
19    line: 'server.document-root = "/usr/local/www/lighttpd"'

```

Example of lineinfile with dict

File `/usr/local/etc/lighttpd/lighttpd.conf` for `lighttpd`

[`contrib/lighttpd/conf-light/files.d/lighttpd-lighttpdconf-dict`]

```

1  lighttpd-lighttpdconf:
2  path: '/usr/local/etc/lighttpd/lighttpd.conf'
3  create: true
4  owner: 'root'
5  group: 'wheel'
6  mode: '0644'
7  handlers:
8  - 'reload lighttpd'
9  assignment: ' = '
10 dict:
11 - {key: 'server.port', value: '"80"'}
12 - {key: 'server.use-ipv6', value: '"disable"'}
13 - {key: 'server.username', value: '"www"'}
14 - {key: 'server.groupname', value: '"www"'}
15 - {key: 'server.document-root', value: '"usr/local/www/lighttpd"'}

```

See Also

See also:

- See [files-lineinfile.yml](#) at GitHub how the files are modified or created by the Ansible module `lineinfile`
- See [files-lineinfile.yml](#) annotated source code

blockinfile

Create or configure blocks in files.

Parameters for blockinfile

Parameter	Type	Comments
path	string required	Path to file
blocks	list required	List of dictionaries (see files-blockinfile.yml)
	marker	string required
	block	string required

owner	string	Owner of the file
group	string	Group of the file
mode	string	Mode of the file
create	boolean	Create if does not exist
validate	string	Command to validate file
handlers	list	List of handlers

Example of blockinfile

Create the description of the file (2) and declare the list of the blocks (10)

[contrib/lighttpd_nagios/conf-light/files.d/lighttpd-modulesconf.yml]

```

1  lighttpd-modulesconf:
2  path: '/usr/local/etc/lighttpd/modules.conf'
3  create: true
4  owner: 'root'
5  group: 'wheel'
6  mode: '0644'
7  copyfile: '{{ cl_lighttpd_modulesconf_copy }}'
8  markers: '{{ cl_lighttpd_modulesconf_markers }}'
9  lines: '{{ cl_lighttpd_modulesconf_lines }}'
10 blocks: '{{ cl_lighttpd_modulesconf_blocks }}'
11 handlers:
12   - 'configtest lighttpd'
13   - 'reload lighttpd'
    
```

Create the list of the blocks `cl_lighttpd_modulesconf_blocks` (89)

[contrib/lighttpd_nagios/cl-lighttpd.yml]

```

84 cl_lighttpd_modulesconf_server_modules:
85   - mod_access
86   - mod_alias
87   - mod_auth
88   - mod_setenv
89 cl_lighttpd_modulesconf_blocks:
90   - marker: 'server.modules'
91     block: |
92         server.modules = (
93             {% for module in cl_lighttpd_modulesconf_server_modules %}
94                 "{{ module }}",
95             {% endfor %}
96         )
    
```

Then, the command

```
shell> ansible-playbook config-light.yml -t cl_files_blockinfile
```

will create this block in `modules.conf`

```
1 ##
2
3 # BEGIN ANSIBLE MANAGED BLOCK server.modules
4 server.modules = (
5     "mod_access",
6     "mod_alias",
7     "mod_auth",
8     "mod_setenv",
9 )
10 # END ANSIBLE MANAGED BLOCK server.modules
11
12 ##
```

See Also

See also:

- See [files-blockinfile.yml](#) at GitHub how the files are modified by the Ansible module `blockinfile`
- See [files-blockinfile.yml](#) annotated source code

ini_file

Create or configure ini entries in a file.

Parameters for ini_file

Parameter	Type	Comments
<code>path</code>	string required	Path to file
<code>ini</code>	list required	<code>ini_file</code> parameters list of dictionaries (see <code>files-inifile.yml</code>)
	<code>section</code>	string required Section name in INI file
	<code>option</code>	string Name of the option
	<code>value</code>	string Value of the option
	<code>state</code>	string If absent the option or section will be removed
	<code>allow_no_value</code>	boolean Allow option without value
	<code>no_extra_spaces</code>	boolean Do not insert spaces
 <TBD>
<code>owner</code>	string	Owner of the file
<code>group</code>	string	Group of the file
<code>mode</code>	string	Mode of the file
<code>attributes</code>	string	Attributes of the file
<code>create</code>	boolean	Create file
<code>handlers</code>	list	List of handlers

Example of ini_file

<TODO: No example yet>

See Also

See also:

- See [files-inifile.yml](#) at GitHub how the files are modified or created by the Ansible module `ini_file`
- See [files-inifile.yml](#) annotated source code

Community General Collection

- [ini_file.py](#)

Note: Quoting [Multiple options with the same name exist #273](#) “There is no “The INI format”, there are many different interpretations out there what valid INI formats should be ...”

2.7 Best practice

2.7.1 Check syntax of setup

Check syntax of the playbook

```
shell> ansible-playbook pb.yml --syntax-check
```

Check syntax of setup and display variables

```
shell> ansible-playbook pb.yml -t cl_setup -e cl_debug=true --check --diff
```

If you want to see the values of the variables enable debug output `cl_debug=true`.

2.7.2 Setup

Collect variables, create handlers, check sanity, and display variables. When you take a look at `tasks/main.yml` you’ll see that the first four groups of the tasks (`setup`, `vars`, `sanity`, and `debug`) are tagged `always`. As a result, when you apply any of the four tags (`cl_setup`, `cl_vars`, `cl_sanity`, and `cl_debug`) all four groups of the tasks, and only these four groups of the tasks, will be executed. All four commands below are equivalent to the command

```
shell> ansible-playbook pb.yml -t cl_setup,cl_vars,cl_sanity,cl_debug
```

- Create variables. Take a look at directory `conf-light/assemble/` what files were created

```
shell> ansible-playbook pb.yml -t cl_vars
```

- Test sanity. Then disable this task `cl_sanity: false` to speedup the playbook

```
shell> ansible-playbook pb.yml -t cl_sanity
```

- Create handlers. Take a look at directory `roles/vbotka.config_light/handlers` what handlers were created. Run this task once to create the handlers. Then disable this task `cl_setup: false` to speedup the playbook

```
shell> ansible-playbook pb.yml -t cl_setup
```

- Display variables. Then disable this task `cl_debug: false` to speedup the playbook

```
shell> ansible-playbook pb.yml -t cl_debug
```

2.7.3 Validation

Create the variables `cl_assemble_validate` and `cl_handlers_validate` if you want to enable validation of the created handlers and assembled data. See `defaults/main.yml`. You'll have to install the package `yamllint`.

2.7.4 Check syntax

Check syntax of the complete playbook

```
shell> ansible-playbook pb.yml --syntax-check
```

2.7.5 Manage packages

Dry-run the management of packages

```
shell> ansible-playbook pb.yml -t cl_packages -e cl_install=true -CD
```

Manage packages

```
shell> ansible-playbook pb.yml -t cl_packages -e cl_install=true
```

Then disable the installation `cl_install=false` to speedup the playbook.

2.7.6 Manage states of files

Dry-run the management of files' states

```
shell> ansible-playbook pb.yml -t cl_states -CD
```

Set the states (existence and attributes) of the files

```
shell> ansible-playbook pb.yml -t cl_states
```

2.7.7 Manage configuration files

Dry-run the configuration of files

```
shell> ansible-playbook pb.yml -t cl_files -CD
```

Create and modify files

```
shell> ansible-playbook pb.yml -t cl_files
```

2.7.8 Manage services

Dry-run the configuration of services

```
shell> ansible-playbook pb.yml -t cl_services -CD
```

Configure services

```
shell> ansible-playbook pb.yml -t cl_services
```

2.7.9 Idempotency

The role and the configuration data in the examples are idempotent. When the application is installed and configured there should be no changes reported by *ansible-playbook* when running the playbook repeatedly. Disable setup, sanity, debug, and install to speedup the execution when running the playbook periodically to audit the configuration

```
shell> ansible-playbook pb.yml -e cl_setup=false \  
      -e cl_sanity=false \  
      -e cl_debug=false \  
      -e cl_install=false
```


Table of Contents

- *Examples*
 - *FreeBSD Postfix*
 - * *Handlers*
 - *contrib/postfix/conf-light/handlers.d/postfix-freebsd.yml*
 - *contrib/postfix/conf-light/handlers.d/sendmail-freebsd.yml*
 - * *Packages*
 - *contrib/postfix/conf-light/packages.d/postfix.yml*
 - * *Services*
 - *contrib/postfix/conf-light/services.d/postfix.yml*
 - *contrib/postfix/conf-light/services.d/sendmail.yml*
 - * *Files*
 - *contrib/postfix/config-light-postfix.yml*
 - *contrib/postfix/conf-light/files.d/mailer-conf.yml*
 - *contrib/postfix/conf-light/files.d/periodic-conf.yml*
 - *contrib/postfix/conf-light/files.d/postfix-main-cf.yml*
 - *contrib/postfix/conf-light/files.d/rc-conf.yml*
 - *Armbian Simple SMTP*
 - * *Packages*
 - *contrib/ssmtp/conf-light/packages.d/ssmtp.yml*

```
* Files
  · contrib/ssmtp/config-light-ssmtp.yml
  · contrib/ssmtp/conf-light/files.d/revaliases.yml
  · contrib/ssmtp/conf-light/files.d/ssmtp-conf.yml
```

3.1 FreeBSD Postfix

3.1.1 Handlers

contrib/postfix/conf-light/handlers.d/postfix-freebsd.yml

Synopsis: Create handlers for Postfix.

Use template (2) to create handlers.

[contrib/postfix/conf-light/handlers.d/postfix-freebsd.yml]

```

1 postfix_freebsd:
2   template: handlers-auto1.yml.j2
3   handlers:
4
5   - handler: 'enable and start postfix'
6     module: service
7     params:
8       - 'name: postfix'
9       - 'state: started'
10      - 'enabled: true'
11
12  - handler: 'disable and stop postfix'
13    module: service
14    params:
15      - 'name: postfix'
16      - 'state: stopped'
17      - 'enabled: false'
18
19  - handler: 'reload postfix'
20    module: service
21    params:
22      - 'name: postfix'
23      - 'state: reloaded'
24    conditions:
25      - '- cl_service_postfix_enable|bool'
26
27  - handler: 'restart postfix'
28    module: service
29    params:
30      - 'name: postfix'
31      - 'state: restarted'
32    conditions:
33      - '- cl_service_postfix_enable|bool'
34
35  - handler: 'postfix check'
36    module: command
```

(continues on next page)

(continued from previous page)

```

37     params:
38         - 'cmd: /usr/local/sbin/postfix check'
39
40     - handler: 'newaliases'
41       module: command
42       params:
43         - 'cmd: /usr/bin/newaliases'
44
45     # - handler: 'postmap smtp sasl passwords'
46       # module: command
47       # params:
48     #   - 'cmd: /usr/local/sbin/postmap {{ postfix_main_cf_smtp_sasl_password_maps }}'
49     ↪
50
51     # - handler: 'postmap virtual aliases'
52       # module: command
53     #   params:
54     #     - cmd: /usr/local/sbin/postmap {{ postfix_virtual }}'

```

See also:

See *setup.yml* how the handlers are created.

contrib/postfix/conf-light/handlers.d/sendmail-freebsd.yml

Synopsis: Create handlers for Sendmail.

Use template (2) to create handlers.

[contrib/postfix/conf-light/handlers.d/sendmail-freebsd.yml]

```

1  sendmail_freebsd:
2  template: handlers-auto1.yml.j2
3  handlers:
4
5  - handler: 'enable and start sendmail'
6    module: service
7    params:
8      - 'name: sendmail'
9      - 'state: started'
10     - 'enabled: true'
11
12 - handler: 'disable and stop sendmail'
13   module: service
14   params:
15     - 'name: sendmail'
16     - 'state: stopped'
17     - 'enabled: false'
18
19 - handler: 'reload sendmail'
20   module: service
21   params:
22     - 'name: sendmail'
23     - 'state: reloaded'
24   conditions:
25     - '- cl_service_sendmail_enable|bool'
26

```

(continues on next page)

(continued from previous page)

```

27 - handler: 'restart sendmail'
28   module: service
29   params:
30     - 'name: sendmail'
31     - 'state: restarted'
32   conditions:
33     - '- cl_service_sendmail_enable|bool'
34
35 - handler: 'start sendmail'
36   module: service
37   params:
38     - 'name: sendmail'
39     - 'state: started'
40
41 - handler: 'stop sendmail'
42   module: service
43   params:
44     - 'name: sendmail'
45     - 'state: stopped'

```

See also:

See *setup.yml* how the handlers are created.

3.1.2 Packages

contrib/postfix/conf-light/packages.d/postfix.yml

Synopsis: Install Postfix.

Use package or port (3) to install Postfix.

[contrib/postfix/conf-light/packages.d/postfix.yml]

```

1 postfix:
2   name:
3     - 'mail/postfix'

```

See also:

See *packages.yml* how the FreeBSD packages or ports are installed.

3.1.3 Services

contrib/postfix/conf-light/services.d/postfix.yml

Synopsis: Configure Postfix service.

Set service (2) state (3). Run the service on boot (4).

[contrib/postfix/conf-light/services.d/postfix.yml]

```

1 postfix:
2   name: 'postfix'
3   state: '{{ cl_service_postfix_state }}'
4   enabled: '{{ cl_service_postfix_enable }}'

```

See also:

See custom Postfix variables *contrib/postfix/config-light-postfix.yml*. See *services.yml* how the services are configured.

contrib/postfix/conf-light/services.d/sendmail.yml

Synopsis: Configure Sendmail service.

Set service (2) state (3). Do not run the service on boot (4).

[contrib/postfix/conf-light/services.d/sendmail.yml]

```

1 sendmail:
2   name: 'sendmail'
3   state: '{{ cl_service_sendmail_state }}'
4   enabled: '{{ cl_service_sendmail_enable }}'
```

See also:

See custom Postfix variables *contrib/postfix/config-light-postfix.yml*.

3.1.4 Files**contrib/postfix/config-light-postfix.yml**

Synopsis: Custom variables for Postfix.

Put the host-specific variables (6) into the `host_vars`. Optionally other variables might be put into the `group_vars`.

[contrib/postfix/config-light-postfix.yml]

```

1 ---
2
3 # 28.4.2. Replace the Default MTA
4 # https://www.freebsd.org/doc/en_US.ISO8859-1/books/handbook/mail-changingmta.html
5
6 cl_myhostname: host99.region9.example.com
7
8 # conf-light/files.d/mailler-conf
9 # Execute the Postfix sendmail program, named /usr/local/sbin/sendmail
10 cl_maillerconf:
11   - 'sendmail      /usr/local/sbin/sendmail'
12   - 'send-mail    /usr/local/sbin/sendmail'
13   - 'mailq        /usr/local/sbin/sendmail'
14   - 'newaliases   /usr/local/sbin/sendmail'
15
16 # conf-light/files.d/rc-rconf
17 cl_rcconf_postfix_enable: 'YES'
18 cl_rcconf_sendmail_enable: 'NO'
19 cl_rcconf_sendmail_submit_enable: 'NO'
20 cl_rcconf_sendmail_outbound_enable: 'NO'
21 cl_rcconf_sendmail_msp_queue_enable: 'NO'
22
23 # conf-light/files.d/periodic-conf
24 cl_periodicconf_daily_clean_hoststat_enable: "NO"
25 cl_periodicconf_daily_status_mail_rejects_enable: "NO"
```

(continues on next page)

(continued from previous page)

```

26 cl_periodicconf_daily_status_include_submit_mailq: "NO"
27 cl_periodicconf_daily_submit_queuerun: "NO"
28
29 # Services
30 cl_service_sendmail_enable: false
31 cl_service_sendmail_state: 'stopped'
32 cl_service_postfix_enable: true
33 cl_service_postfix_state: 'started'
34
35 # If you are using SASL, you need to make sure that postfix has access
36 # to read the sasldb file. This is accomplished by adding postfix to
37 # group mail and making the /usr/local/etc/sasl* file(s) readable by
38 # group mail (this should be the default for new installs).
39
40 # EOF
41 ...

```

contrib/postfix/conf-light/files.d/mailler-conf.yml

Synopsis: Create file.

Create file (2) from the template (4).

[contrib/postfix/conf-light/files.d/mailler-conf.yml]

```

1 mailerconf:
2   path: '/etc/mail/mailler.conf'
3   template:
4     path: 'mailler.conf.j2'
5     force: true
6     owner: 'root'
7     group: 'wheel'
8     mode: '0644'

```

contrib/postfix/conf-light/files.d/periodic-conf.yml

Synopsis: Modify file.

Modify file (2) with the lines (7).

[contrib/postfix/conf-light/files.d/periodic-conf.yml]

```

1 periodic_conf:
2   path: '/etc/periodic.conf'
3   create: true
4   owner: 'root'
5   group: 'wheel'
6   mode: '0644'
7   lines:
8     - regexp: '^daily_clean_hoststat_enable(.*)$'
9       line: 'daily_clean_hoststat_enable="{{ cl_periodicconf_daily_clean_hoststat_
↵enable }}"'
10    - regexp: '^daily_status_mail_rejects_enable(.*)$'
11      line: 'daily_status_mail_rejects_enable="{{ cl_periodicconf_daily_status_mail_
↵rejects_enable }}"'

```

(continues on next page)

(continued from previous page)

```

12 - regexp: '^daily_status_include_submit_mailq(.*)$'
13   line: 'daily_status_include_submit_mailq="{{ cl_periodicconf_daily_status_
↵include_submit_mailq }}"'
14 - regexp: '^daily_submit_queuerun(.*)$'
15   line: 'daily_submit_queuerun="{{ cl_periodicconf_daily_submit_queuerun }}"'

```

contrib/postfix/conf-light/files.d/postfix-main-cf.yml

Synopsis: Modify file and notify handlers.

Modify file (2) with the lines (9) and notify handlers (7).

[contrib/postfix/conf-light/files.d/postfix-main-cf.yml]

```

1 postfix_main_cf:
2   path: '/usr/local/etc/postfix/main.cf'
3   create: true
4   owner: 'root'
5   group: 'wheel'
6   mode: '0644'
7   handlers:
8     - 'postfix_freebsd reload postfix'
9   lines:
10    - regexp: '^myhostname\s*=\s*(.*)$'
11      line: 'myhostname = {{ cl_myhostname }}'

```

contrib/postfix/conf-light/files.d/rc-conf.yml

Synopsis: Modify file.

Modify file (2) with the lines (7).

[contrib/postfix/conf-light/files.d/rc-conf.yml]

```

1 rcconf:
2   path: '/etc/rc.conf'
3   create: true
4   owner: 'root'
5   group: 'wheel'
6   mode: '0644'
7   lines:
8     - regexp: '^sendmail_enable(.*)$'
9       line: 'sendmail_enable="{{ cl_rcconf_sendmail_enable }}"'
10    - regexp: '^sendmail_submit_enable(.*)$'
11      line: 'sendmail_submit_enable="{{ cl_rcconf_sendmail_submit_enable }}"'
12    - regexp: '^sendmail_outbound_enable(.*)$'
13      line: 'sendmail_outbound_enable="{{ cl_rcconf_sendmail_outbound_enable }}"'
14    - regexp: '^sendmail_msp_queue_enable(.*)$'
15      line: 'sendmail_msp_queue_enable="{{ cl_rcconf_sendmail_msp_queue_enable }}"'
16    - regexp: '^postfix_enable(.*)$'
17      line: 'postfix_enable="{{ cl_rcconf_postfix_enable }}"'

```

3.2 Armbian Simple SMTP

3.2.1 Packages

contrib/ssmtp/conf-light/packages.d/ssmtp.yml

Synopsis: Install Simple SMTP.

Use package (3) to install sSMTP.

[contrib/ssmtp/conf-light/packages.d/ssmtp.yml]

```

1 ssmtp:
2   name:
3     - 'ssmtp'
```

See also:

See *packages.yml* how the Linux packages are installed.

3.2.2 Files

contrib/ssmtp/config-light-ssmtp.yml

Synopsis: Custom variables for sSMTP.

Put the host-specific variables (7) into the `host_vars`. Optionally other variables might be put into the `group_vars`.

[contrib/ssmtp/config-light-ssmtp.yml]

```

1 ---
2
3 # sSMTP - Simple SMTP
4 # https://wiki.debian.org/sSMTP
5
6 # linux-postinstall FQDN
7 lp_fqdn: "host99.region9.example.com"
8
9 # NEVER USE PLAINTEXT PASSWORD. USE VAULT INSTEAD
10 smtp_client_password_mail_example_com: "PASSWORD"
11
12 # conf-light/files.d/ssmtp-conf
13 cl_ssmtp_srv: "mail.example.com"
14 cl_ssmtp_srv_domain: "example.com"
15
16 cl_ssmtp_postmaster_address: "postmaster@{{ lp_fqdn }}"
17 cl_ssmtp_mailhub: "{{ cl_ssmtp_srv }}:587"
18 cl_ssmtp_rewriteDomain: "{{ cl_ssmtp_srv_domain }}"
19
20 cl_ssmtp_usetls: "Yes"
21 cl_ssmtp_usestarttls: "Yes"
22
23 cl_ssmtp_authuser: "smtp_client"
24 cl_ssmtp_authpass: "{{ smtp_client_password_mail_example_com }}"
25 cl_ssmtp_authmethod: "LOGIN"
```

(continues on next page)

(continued from previous page)

```

26
27 cl_ssmtp_FromLineOverride: "yes"
28
29 # conf-light/files.d/revaliases
30 cl_ssmtp_revaliases:
31 - "root:root@{{ cl_ssmtp_srv_domain }}:{{ cl_ssmtp_srv }}:587"
32 - "admin:admin@{{ cl_ssmtp_srv_domain }}:{{ cl_ssmtp_srv }}:587"
33 - "user1:user1@{{ cl_ssmtp_srv_domain }}:{{ cl_ssmtp_srv }}:587"
34
35 # EOF
36 ...

```

contrib/ssmtp/conf-light/files.d/revaliases.yml

Synopsis: Create file.

Create file (2) from the template (8).

[contrib/ssmtp/conf-light/files.d/revaliases.yml]

```

1 revaliases:
2 path: '/etc/ssmtp/revaliases'
3 force: true
4 owner: 'root'
5 group: 'mail'
6 mode: 'u=rw,g=r'
7 template:
8 path: 'revaliases.j2'

```

See also:

See template `revaliases.j2`. See how files are created from template `files-template.yml`.

contrib/ssmtp/conf-light/files.d/ssmtp-conf.yml

Synopsis: Create file.

Create file (2) from the template (8).

[contrib/ssmtp/conf-light/files.d/ssmtp-conf.yml]

```

1 ssmtp_conf:
2 path: '/etc/ssmtp/ssmtp.conf'
3 force: true
4 owner: 'root'
5 group: 'mail'
6 mode: 'u=rw,g=r'
7 template:
8 path: 'ssmtp.conf.j2'

```

See also:

See template `ssmtp.conf.j2`. See how files are created from template `files-template.yml`.

4.1 Tasks

4.1.1 main.yml

Synopsis: Main task.

Import tasks if enabled.

[tasks/main.yml]

```
1 ---
2 # tasks for config_light
3
4 - name: Import setup.yml
5   ansible.builtin.import_tasks: setup.yml
6   when: cl_setup|bool
7   delegate_to: localhost
8   run_once: true
9   tags: [cl_setup, always]
10
11 - name: Import vars.yml
12   ansible.builtin.import_tasks: vars.yml
13   tags: [cl_vars, always]
14
15 - name: Import sanity.yml
16   ansible.builtin.import_tasks: sanity.yml
17   when: cl_sanity|bool
18   tags: [cl_sanity, always]
19
20 - name: Import debug.yml
21   ansible.builtin.import_tasks: debug.yml
22   when: cl_debug|bool
23   tags: cl_debug
```

(continues on next page)

(continued from previous page)

```

24
25 - name: Import packages.yml
26   ansible.builtin.import_tasks: packages.yml
27   when: cl_install|bool
28   tags: cl_packages
29
30 - name: Import states.yml
31   ansible.builtin.import_tasks: states.yml
32   tags: cl_states
33
34 - name: Import files.yml
35   ansible.builtin.import_tasks: files.yml
36   tags: cl_files
37
38 - name: Import services.yml
39   ansible.builtin.import_tasks: services.yml
40   tags: cl_services
41
42 # EOF
43 ...

```

4.1.2 setup.yml

Synopsis: Configure setup.

Description of the task.

[tasks/setup.yml]

```

1 ---
2
3 - name: "setup: Debug"
4   vars:
5     msg: |-
6       cl_dird [{{ cl_dird }}]
7       cl_dird_owner [{{ cl_dira_owner|default('UNDEFINED') }}]
8       cl_dird_group [{{ cl_dira_group|default('UNDEFINED') }}]
9       cl_dird_dmode [{{ cl_dira_dmode }}]
10      cl_handlersd_dir [{{ cl_handlersd_dir }}]
11      cl_packagesd_dir [{{ cl_packagesd_dir }}]
12      cl_servicesd_dir [{{ cl_servicesd_dir }}]
13      cl_filesd_dir [{{ cl_filesd_dir }}]
14      cl_statesd_dir [{{ cl_statesd_dir }}]
15      cl_dira [{{ cl_dira }}]
16      cl_dira_owner [{{ cl_dira_owner|default('UNDEFINED') }}]
17      cl_dira_group [{{ cl_dira_group|default('UNDEFINED') }}]
18      cl_dira_dmode [{{ cl_dira_dmode }}]
19      cl_dira_fmode [{{ cl_dira_fmode }}]
20      cl_assemble_regexp [{{ cl_assemble_regexp }}]
21      cl_handlersd [{{ cl_handlersd }}]
22      cl_packagesd [{{ cl_packagesd }}]
23      cl_servicesd [{{ cl_servicesd }}]
24      cl_filesd [{{ cl_filesd }}]
25      cl_statesd [{{ cl_statesd }}]
26      cl_handlers_clean_all [{{ cl_handlers_clean_all }}]
27      cl_handlers_create [{{ cl_handlers_create }}]

```

(continues on next page)

(continued from previous page)

```

28     cl_handlers_delete [{{ cl_handlers_delete }}]
29     cl_handlers_dir_become [{{ cl_handlers_dir_become }}]
30     cl_handlers_dir_owner [{{ cl_handlers_dir_owner|default('UNDEFINED') }}]
31     cl_handlers_dir_group [{{ cl_handlers_dir_group|default('UNDEFINED') }}]
32     cl_handlers_dir_dmode [{{ cl_handlers_dir_dmode|default('UNDEFINED') }}]
33     cl_handlers_main_mode [{{ cl_handlers_main_mode|default('UNDEFINED') }}]
34 ansible.builtin.debug:
35     msg: "{{ '{ }'.format(msg) }}"
36 when: cl_debug|bool
37
38 # directories
39 - name: "setup: Create directories in {{ cl_dird }}"
40 ansible.builtin.file:
41     state: directory
42     path: "{{ item }}"
43     owner: "{{ cl_dird_owner|default(omit) }}"
44     group: "{{ cl_dird_group|default(omit) }}"
45     mode: "{{ cl_dird_dmode }}"
46 loop:
47     - "{{ cl_dird }}"
48     - "{{ cl_handlersd_dir }}"
49     - "{{ cl_packagesd_dir }}"
50     - "{{ cl_servicesd_dir }}"
51     - "{{ cl_filesd_dir }}"
52     - "{{ cl_statesd_dir }}"
53
54 - name: "setup: Create directory {{ cl_dira }}"
55 ansible.builtin.file:
56     state: directory
57     path: "{{ cl_dira }}"
58     owner: "{{ cl_dira_owner|default(omit) }}"
59     group: "{{ cl_dira_group|default(omit) }}"
60     mode: "{{ cl_dira_dmode }}"
61
62 # handlers
63 - name: "setup: Create dir {{ role_path }}/handlers"
64 become: "{{ cl_handlers_dir_become }}"
65 ansible.builtin.file:
66     state: directory
67     path: "{{ role_path }}/handlers"
68     owner: "{{ cl_handlers_dir_owner|default(omit) }}"
69     group: "{{ cl_handlers_dir_group|default(omit) }}"
70     mode: "{{ cl_handlers_dir_dmode|default(omit) }}"
71
72 - name: "setup: Create handlers/main.yml"
73 ansible.builtin.lineinfile:
74     path: "{{ role_path }}/handlers/main.yml"
75     insertbefore: BOF
76     line: "---"
77     validate: "{{ cl_handlers_validate|default(omit) }}"
78     backup: "{{ cl_backup }}"
79     create: true
80     mode: "{{ cl_handlers_main_mode|default(omit) }}"
81
82 - name: "setup: Assemble handlers"
83 ansible.builtin.import_tasks: vars-handlers.yml
84

```

(continues on next page)

(continued from previous page)

```

85 - name: "setup: Delete all handlers"
86   block:
87     - name: "setup: Find all handlers"
88       ansible.builtin.find:
89         path: "{{ role_path }}/handlers"
90         patterns: handlers-auto-*.yaml
91         register: result
92     - name: "setup: Debug result"
93       ansible.builtin.debug:
94         var: result
95     when: cl_debug|bool
96   - name: "setup: Delete all handlers"
97     ansible.builtin.file:
98       state: absent
99       path: "{{ item.path }}"
100      loop: "{{ result.files }}"
101      loop_control:
102        label: "{{ item.path|basename }}"
103   - name: "setup: Exclude all found handlers from handlers/main.yml"
104     ansible.builtin.lineinfile:
105       state: absent
106       path: "{{ role_path }}/handlers/main.yml"
107       line: "- import_tasks: {{ item.path|basename }}"
108       backup: "{{ cl_backup }}"
109       loop: "{{ result.files }}"
110       loop_control:
111         label: "{{ item.path|basename }}"
112   when: cl_handlers_clean_all|bool
113
114 - name: "setup: Delete handlers listed in cl_handlers"
115   block:
116     - name: "setup: Delete handlers listed in cl_handlers"
117       ansible.builtin.file:
118         state: absent
119         path: "{{ role_path }}/handlers/handlers-auto-{{ item.key }}.yaml"
120         loop: "{{ cl_handlers|dict2items }}"
121         loop_control:
122           label: "{{ role_path }}/handlers/handlers-auto-{{ item.key }}.yaml"
123     - name: "setup: Exclude handlers from handlers/main.yml"
124       ansible.builtin.lineinfile:
125         state: absent
126         path: "{{ role_path }}/handlers/main.yml"
127         line: "- import_tasks: handlers-auto-{{ item.key }}.yaml"
128         backup: "{{ cl_backup }}"
129         loop: "{{ cl_handlers|dict2items }}"
130         loop_control:
131           label: "{{ role_path }}/handlers/handlers-auto-{{ item.key }}.yaml"
132   when: cl_handlers_delete|bool
133
134 - name: "setup: Create handlers listed in cl_handlers"
135   block:
136     - name: "setup: Create handlers listed in cl_handlers"
137       ansible.builtin.template:
138         dest: "{{ role_path }}/handlers/handlers-auto-{{ item.key }}.yaml"
139         src: "{{ item.value.template }}"
140         mode: "{{ item.value.mode|default(omit) }}"
141         validate: "{{ cl_handlers_validate|default(omit) }}"

```

(continues on next page)

(continued from previous page)

```

142     backup: "{{ cl_backup }}"
143     loop: "{{ cl_handlers|dict2items }}"
144     loop_control:
145       label: "{{ role_path }}/handlers/handlers-auto-{{ item.key }}.yaml"
146   - name: "setup: Include handlers in handlers/main.yaml"
147     ansible.builtin.lineinfile:
148       path: "{{ role_path }}/handlers/main.yaml"
149       line: "- import_tasks: handlers-auto-{{ item.key }}.yaml"
150       validate: "{{ cl_handlers_validate|default(omit) }}"
151       backup: "{{ cl_backup }}"
152     loop: "{{ cl_handlers|dict2items }}"
153     loop_control:
154       label: "{{ role_path }}/handlers/handlers-auto-{{ item.key }}.yaml"
155   when: cl_handlers_create|bool
156
157 # EOF
158 ...

```

4.1.3 vars-handlers.yml

Synopsis: Configure vars-handlers.

Description of the task.

[tasks/vars-handlers.yml]

```

1 ---
2
3 - block:
4   - name: "vars-handlers: Assemble handlers to {{ cl_handlersd }}"
5     ansible.builtin.assemble:
6       regexp: "{{ cl_assemble_regexp }}"
7       src: "{{ cl_handlersd_dir }}"
8       dest: "{{ cl_handlersd }}"
9       owner: "{{ cl_dira_owner|default(omit) }}"
10      group: "{{ cl_dira_group|default(omit) }}"
11      mode: "{{ cl_dira_fmode }}"
12      validate: "{{ cl_assemble_validate|default(omit) }}"
13  rescue:
14    - name: "vars-handlers: Assemble handlers to {{ cl_handlersd }} failed"
15      ansible.builtin.debug:
16        msg: |-
17          Can not assemble handlers. End of play.
18          {{ ansible_failed_result }}
19    - name: "vars-handlers: End of play"
20      meta: end_play
21
22 - name: "vars-handlers: Include files from {{ cl_handlersd }} to cl_handlersd_items"
23   ansible.builtin.include_vars:
24     file: "{{ cl_handlersd }}"
25     name: cl_handlersd_items
26
27 - name: "vars-handlers: Combine cl_handlers with cl_handlersd_items"
28   ansible.builtin.set_fact:
29     cl_handlers: "{{ cl_handlers|combine(cl_handlersd_items|default({})) }}"
30

```

(continues on next page)

(continued from previous page)

```

31 - name: "vars-handlers: Debug"
32   ansible.builtin.debug:
33     var: cl_handlers
34     when: cl_debug|bool
35
36 # EOF
37 ...

```

4.1.4 vars.yml

Synopsis: Configure vars.

Description of the task.

[tasks/vars.yml]

```

1 ---
2
3 - name: "vars: Debug"
4   vars:
5     msg: |-
6       cl_handlersd_dir [{{ cl_handlersd_dir }}]
7       cl_packagesd_dir [{{ cl_packagesd_dir }}]
8       cl_statesd_dir [{{ cl_statesd_dir }}]
9       cl_servicesd_dir [{{ cl_servicesd_dir }}]
10      cl_filesd_dir [{{ cl_filesd_dir }}]
11      cl_dira [{{ cl_dira }}]
12      cl_handlersd [{{ cl_handlersd }}]
13      cl_packagesd [{{ cl_packagesd }}]
14      cl_statesd [{{ cl_statesd }}]
15      cl_servicesd [{{ cl_servicesd }}]
16      cl_filesd [{{ cl_filesd }}]
17   ansible.builtin.debug:
18     msg: "{{ '{ }'.format(msg) }}"
19     when: cl_debug|bool
20
21 - name: "vars: Packages"
22   ansible.builtin.import_tasks: vars-packages.yml
23
24 - name: "vars: States"
25   ansible.builtin.import_tasks: vars-states.yml
26
27 - name: "vars: Services"
28   ansible.builtin.import_tasks: vars-services.yml
29
30 - name: "vars: Files"
31   ansible.builtin.import_tasks: vars-files.yml
32
33 # EOF
34 ...

```

4.1.5 vars-packages.yml

Synopsis: Configure vars-packages.

Description of the task.

[tasks/vars-packages.yml]

```

1 ---
2
3 - block:
4   - name: "vars-packages: Assemble packages to {{ cl_packagesd }}"
5     ansible.builtin.assemble:
6       regexp: "{{ cl_assemble_regexp }}"
7       src: "{{ cl_packagesd_dir }}"
8       dest: "{{ cl_packagesd }}"
9       owner: "{{ cl_dira_owner|default(omit) }}"
10      group: "{{ cl_dira_group|default(omit) }}"
11      mode: "{{ cl_dira_fmode }}"
12      validate: "{{ cl_assemble_validate|default(omit) }}"
13      delegate_to: localhost
14    rescue:
15      - name: "vars-packages: Assemble packages to {{ cl_packagesd }} failed"
16        ansible.builtin.debug:
17          msg: |-
18            Can not assemble packages. End of play.
19            {{ ansible_failed_result }}
20      - name: "vars-packages: End of play"
21        meta: end_play
22
23 - name: "vars-packages: Include files from {{ cl_packagesd }} to cl_packagesd_items"
24   ansible.builtin.include_vars:
25     file: "{{ cl_packagesd }}"
26     name: cl_packagesd_items
27
28 - name: "vars-packages: Combine cl_packages with cl_packagesd_items"
29   ansible.builtin.set_fact:
30     cl_packages: "{{ cl_packages|combine(cl_packagesd_items|default({})) }}"
31
32 - name: "vars-packages: Debug"
33   ansible.builtin.debug:
34     var: cl_packages
35     when: cl_debug|bool
36
37 # TODO: sort|unique list of packages/ports
38
39 # EOF
40 ...

```

4.1.6 vars-states.yml

Synopsis: Configure vars-states.

Description of the task.

[tasks/vars-states.yml]

```

1 ---
2
3 - block:
4   - name: "vars-states: Assemble states to {{ cl_states }}"

```

(continues on next page)

(continued from previous page)

```

5  ansible.builtin.assemble:
6      regexp: "{{ cl_assemble_regexp }}"
7      src: "{{ cl_statesd_dir }}"
8      dest: "{{ cl_statesd }}"
9      owner: "{{ cl_dira_owner|default(omit) }}"
10     group: "{{ cl_dira_group|default(omit) }}"
11     mode: "{{ cl_dira_fmode }}"
12     validate: "{{ cl_assemble_validate|default(omit) }}"
13     delegate_to: localhost
14 rescue:
15     - name: "vars-states: Assemble states to {{ cl_states }} failed"
16       ansible.builtin.debug:
17         msg: |-
18             Can not assemble states. End of play.
19             {{ ansible_failed_result }}
20     - name: "vars-states: End of play"
21       meta: end_play
22
23 - name: "vars-states: Include files from {{ cl_statesd }} to cl_statesd_items"
24   ansible.builtin.include_vars:
25     file: "{{ cl_statesd }}"
26     name: cl_statesd_items
27
28 - name: "vars-states: Combine cl_states with cl_statesd_items"
29   ansible.builtin.set_fact:
30     cl_states: "{{ cl_states|combine(cl_statesd_items|default({})) }}"
31
32 - name: "vars-states: Debug"
33   ansible.builtin.debug:
34     var: cl_states
35     when: cl_debug|bool
36
37 # EOF
38 ...

```

4.1.7 vars-services.yml

Synopsis: Configure vars-services.

Description of the task.

[tasks/vars-services.yml]

```

1  ---
2
3  - block:
4      - name: "vars-services: Assemble services to {{ cl_servicesd }}"
5        ansible.builtin.assemble:
6          regexp: "{{ cl_assemble_regexp }}"
7          src: "{{ cl_servicesd_dir }}"
8          dest: "{{ cl_servicesd }}"
9          owner: "{{ cl_dira_owner|default(omit) }}"
10         group: "{{ cl_dira_group|default(omit) }}"
11         mode: "{{ cl_dira_fmode }}"
12         validate: "{{ cl_assemble_validate|default(omit) }}"
13         delegate_to: localhost

```

(continues on next page)

(continued from previous page)

```

14  rescue:
15      - name: "vars-services: Assemble services to {{ cl_servicesd }} failed"
16        ansible.builtin.debug:
17          msg: |-
18            Can not assemble services. End of play.
19            {{ ansible_failed_result }}
20      - name: "vars-services: End of play"
21        meta: end_play
22
23  - name: "vars-services: Include files from {{ cl_servicesd }} to cl_cervicesd_items"
24    ansible.builtin.include_vars:
25      file: "{{ cl_servicesd }}"
26      name: cl_servicesd_items
27
28  - name: "vars-services: Combine cl_services with cl_servicesd_items"
29    ansible.builtin.set_fact:
30      cl_services: "{{ cl_services|combine(cl_servicesd_items|default({})) }}"
31
32  - name: "vars-services: Debug"
33    ansible.builtin.debug:
34      var: cl_services
35    when: cl_debug|bool
36
37  # EOF
38  ...

```

4.1.8 vars-files.yml

Synopsis: Configure vars-files.

Description of the task.

[tasks/vars-files.yml]

```

1  ---
2
3  - block:
4      - name: "vars-files: Assemble files to {{ cl_filesd }}"
5        ansible.builtin.assemble:
6          regexp: "{{ cl_assemble_regexp }}"
7          src: "{{ cl_filesd_dir }}"
8          dest: "{{ cl_filesd }}"
9          owner: "{{ cl_dira_owner|default(omit) }}"
10         group: "{{ cl_dira_group|default(omit) }}"
11         mode: "{{ cl_dira_fmode }}"
12         validate: "{{ cl_assemble_validate|default(omit) }}"
13         delegate_to: localhost
14     rescue:
15         - name: "vars-files: Assemble files to {{ cl_filesd }} failed"
16           ansible.builtin.debug:
17             msg: |-
18               Can not assemble files. End of play.
19               {{ ansible_failed_result }}
20         - name: "vars-files: End of play"
21           meta: end_play
22

```

(continues on next page)

(continued from previous page)

```

23 - name: "vars-files: Include files from {{ cl_filesd }} to cl_filesd_items"
24   ansible.builtin.include_vars:
25     file: "{{ cl_filesd }}"
26     name: cl_filesd_items
27
28 - name: "vars-files: Combine cl_files with cl_filesd_items"
29   ansible.builtin.set_fact:
30     cl_files: "{{ cl_files|combine(cl_filesd_items|default({})) }}"
31
32 - name: "vars-files: Debug"
33   ansible.builtin.debug:
34     var: cl_files
35     when: cl_debug|bool
36
37 # EOF
38 ...

```

4.1.9 sanity.yml

Synopsis: Configure sanity.

Description of the task.

[tasks/sanity.yml]

```

1 ---
2
3 - block:
4   - name: "sanity: Directories to assemble data from must exist"
5     ansible.builtin.debug:
6       msg: "{{ ' {}'.format(msg) }}"
7     vars:
8       msg: |-
9         Directories to assemble data from do not exist. End of play.
10        Hint: Double check existence of the directories
11        {{ cl_handlersd_dir }}
12        {{ cl_packagesd_dir }}
13        {{ cl_statesd_dir }}
14        {{ cl_servicesd_dir }}
15        {{ cl_filesd_dir }}
16   - name: "sanity: End of play"
17     meta: end_play
18   when:
19     - cl_handlersd_dir is not exists or
20       cl_packagesd_dir is not exists or
21       cl_statesd_dir is not exists or
22       cl_servicesd_dir is not exists or
23       cl_filesd_dir is not exists
24
25 - block:
26   - name: "sanity: Check mode not possible without assembled data"
27     ansible.builtin.debug:
28       msg: "{{ ' {}'.format(msg) }}"
29     vars:
30       msg: |-
31         Check mode not possible without assembled data. End of play.

```

(continues on next page)

(continued from previous page)

```

32     Hint: Assemble the variables first.
33     Run: ansible-playbook playbook.yml -t cl_vars
34     - name: "sanity: End of play"
35       ansible.builtin.meta: end_play
36   when:
37     - ansible_check_mode
38     - cl_packagesd is not exists or
39       cl_statesd is not exists or
40       cl_serviced is not exists or
41       cl_filesd is not exists
42
43 - block:
44   - name: "sanity: yamllint must be installed"
45     ansible.builtin.assert:
46       that: _out is match(_regex)
47       fail_msg: |-
48         Failed. yamllint not installed on controller.
49       success_msg: |-
50         Passed. yamllint found on controller.
51     vars:
52       _regex: '^yamllint \d+\.\d+\.\d+$'
53       _out: "{{ lookup('pipe', cl_yamllint ~ ' --version') }}"
54   rescue:
55     - name: "sanity: Rescue: yamllint must be installed"
56       ansible.builtin.debug:
57         msg: "{{ ' {}'.format(msg) }}"
58       vars:
59         msg: |-
60           Failed. yamllint must be installed on controller. {{ cl_yamllint }} does_
↪not exist. End of play.
61     - name: "sanity: Rescue: End of play"
62       ansible.builtin.meta: end_play
63   when: cl_yamllint_missing_fatal|bool
64
65 # EOF
66 ...

```

4.1.10 debug.yml

Synopsis: Configure debug.

Description of the task.

[tasks/debug.yml]

```

1 ---
2
3 - name: "debug: Config Light"
4   vars:
5     msg: |-
6       ansible_os_family [{{ ansible_os_family }}]
7       ansible_distribution [{{ ansible_distribution }}]
8       ansible_distribution_major_version [{{ ansible_distribution_major_version }}]
9       ansible_distribution_version [{{ ansible_distribution_version }}]
10      ansible_distribution_release [{{ ansible_distribution_release }}]
11      ansible_python_version [{{ ansible_python_version }}]

```

(continues on next page)

(continued from previous page)

```
12
13     cl_sanity [{{ cl_sanity }}]
14     cl_setup [{{ cl_setup }}]
15     cl_install [{{ cl_install }}]
16     cl_backup [{{ cl_backup }}]
17     cl_copyfile_delete [{{ cl_copyfile_delete }}]
18     cl_template_delete [{{ cl_template_delete }}]
19
20     cl_handlers_clean_all [{{ cl_handlers_clean_all }}]
21     cl_handlers_delete [{{ cl_handlers_delete }}]
22     cl_handlers_create [{{ cl_handlers_create }}]
23     cl_handlers
24     {{ cl_handlers|to_nice_yaml }}
25     cl_packages
26     {{ cl_packages|to_nice_yaml }}
27     cl_services
28     {{ cl_services|to_nice_yaml }}
29     cl_files
30     {{ cl_files|to_nice_yaml }}
31     cl_states
32     {{ cl_states|to_nice_yaml }}
33
34     cl_dird [{{ cl_dird }}]
35     cl_dird_owner [{{ cl_dird_owner|default('UNDEFINED') }}]
36     cl_dird_group [{{ cl_dird_group|default('UNDEFINED') }}]
37     cl_dird_dmode [{{ cl_dird_dmode }}]
38
39     cl_handlersd_dir [{{ cl_handlersd_dir }}]
40     cl_packagesd_dir [{{ cl_packagesd_dir }}]
41     cl_servicesd_dir [{{ cl_servicesd_dir }}]
42     cl_filesd_dir [{{ cl_filesd_dir }}]
43     cl_statesd_dir [{{ cl_statesd_dir }}]
44
45     cl_dira [{{ cl_dira }}]
46     cl_dira_owner [{{ cl_dira_owner|default('UNDEFINED') }}]
47     cl_dira_group [{{ cl_dira_group|default('UNDEFINED') }}]
48     cl_dira_dmode [{{ cl_dira_dmode }}]
49     cl_dira_fmode [{{ cl_dira_fmode }}]
50     cl_assemble_regexp [{{ cl_assemble_regexp }}]
51
52     cl_handlersd [{{ cl_handlersd }}]
53     cl_packagesd [{{ cl_packagesd }}]
54     cl_servicesd [{{ cl_servicesd }}]
55     cl_filesd [{{ cl_filesd }}]
56     cl_statesd [{{ cl_statesd }}]
57
58     cl_yamlllint_missing_fatal [{{ cl_yamlllint_missing_fatal }}]
59     cl_yamlllint [{{ cl_yamlllint }}]
60     cl_assemble_validate [{{ cl_assemble_validate|default('UNDEFINED') }}]
61     cl_handlers_validate [{{ cl_handlers_validate|default('UNDEFINED') }}]
62
63     cl_files_order
64     {{ cl_files_order|to_yaml }}
65
66     install_retries [{{ install_retries }}]
67     install_delay [{{ install_delay }}]
68
```

(continues on next page)

(continued from previous page)

```

69     frebsd_install_method [{{ frebsd_install_method }}]
70     frebsd_use_packages [{{ frebsd_use_packages }}]
71     cl_services_frebsd_rcconf_auto [{{ cl_services_frebsd_rcconf_auto }}]
72
73     ansible.builtin.debug:
74         msg: "{{ '{ }'.format(msg) }}"
75
76     # EOF
77     ...

```

4.1.11 packages.yml

Synopsis: Configure packages.

Description of the task.

[tasks/packages.yml]

```

1  ---
2
3  - name: "packages: Debug"
4    ansible.builtin.debug:
5        msg: "{{ cl_packages.values()|list }}"
6        when: cl_debug|bool
7
8  # FreeBSD packages - - - - -
9  - block:
10     - name: "packages: Manage FreeBSD packages"
11       community.general.pkgng:
12           name: "{{ item.name }}" # list / elements=string / required
13           state: "{{ item.state|default(omit) }}"
14           annotation: "{{ item.annotation|default(omit) }}"
15           autoremove: "{{ frebsd_pkgng_autoremove|default(omit) }}"
16           cached: "{{ frebsd_pkgng_cached|default(omit) }}"
17           chroot: "{{ frebsd_pkgng_chroot|default(omit) }}"
18           ignore_osver: "{{ frebsd_pkgng_ignore_osver|default(omit) }}"
19           jail: "{{ frebsd_pkgng_jail|default(omit) }}"
20           pkgsite: "{{ frebsd_pkgng_pkgsite|default(omit) }}"
21           rootdir: "{{ frebsd_pkgng_rootdir|default(omit) }}"
22           loop: "{{ cl_packages.values()|list }}"
23           loop_control:
24               label: "{{ item.name }}"
25           register: result
26           until: result is succeeded
27           retries: "{{ install_retries }}"
28           delay: "{{ install_delay }}"
29     - name: "packages: Debug FreeBSD packages"
30       ansible.builtin.debug:
31           var: result
32       when: cl_debug|bool
33   when:
34       - ansible_os_family == "FreeBSD"
35       - frebsd_install_method|lower == "packages"
36
37 # FreeBSD ports - - - - -
38 - block:

```

(continues on next page)

(continued from previous page)

```

39 - name: "packages: Manage FreeBSD ports"
40   community.general.portinstall:
41     name: "{{ item.1 }}" # string / required
42     state: "{{ item.0.state|default(omit) }}"
43     use_packages: "{{ item.0.use_packages|default(freebsd_use_packages) }}"
44   with_subelements:
45     - "{{ cl_packages.values()|list }}"
46     - name
47   loop_control:
48     label: "{{ item.1 }}"
49   register: result
50   until: result is succeeded
51   retries: "{{ install_retries }}"
52   delay: "{{ install_delay }}"
53 - name: "packages: Debug FreeBSD ports"
54   ansible.builtin.debug:
55     var: result
56   when: cl_debug|bool
57 when:
58 - ansible_os_family == "FreeBSD"
59 - freebsd_install_method|lower == "ports"
60
61 # Linux snap - - - - -
62 - block:
63   - name: "packages: Find snap"
64     ansible.builtin.find:
65       paths: "{{ cl_snap_paths }}"
66       patterns: "{{ cl_snap_patterns }}"
67     register: _find_snap
68   - name: "packages: Debug find snap"
69     ansible.builtin.debug:
70       var: _find_snap
71     when: cl_debug|bool
72   - name: "packages: Fail if snap is missing"
73     ansible.builtin.fail:
74       msg: "[ERROR] snap not installed."
75     when: _find_snap.matched == 0
76 vars:
77   _cl_packages_snap: "{{ cl_packages.values()|list|json_query('[?module == `snap`
↪') ]}'"
78 when:
79 - _cl_packages_snap|length > 0
80 - ansible_os_family == "RedHat" or
81   ansible_os_family == "Debian"
82 - block:
83   - name: "packages: Manage Linux packages by snap module"
84     community.general.snap:
85       name: "{{ item.name }}" # list / elements=string
86       state: "{{ item.state|default(omit) }}"
87       loop: "{{ _cl_packages_snap }}"
88     loop_control:
89       label: "{{ item.name }}"
90     register: result
91     until: result is succeeded
92     retries: "{{ install_retries }}"
93     delay: "{{ install_delay }}"
94     when: item.module|default('package') == 'snap'

```

(continues on next page)

(continued from previous page)

```

95 - name: "packages: Debug Linux snap"
96   ansible.builtin.debug:
97     var: result
98     when: cl_debug|bool
99   vars:
100     _cl_packages_snap: "{{ cl_packages.values()|list|json_query('[?module == `snap`'
101     ↪') ] }}"
102   when:
103     - _cl_packages_snap|length > 0
104     - ansible_os_family == "RedHat" or
105       ansible_os_family == "Debian"
106 # Linux package - - - - -
107 - block:
108   - name: "packages: Manage Linux packages by package module"
109     ansible.builtin.package:
110       name: "{{ item.1 }}" # string / required
111       state: "{{ item.0.state|default('present') }}"
112       use: "{{ item.0.use|default('auto') }}"
113     with_subelements:
114       - "{{ cl_packages.values()|list }}"
115       - name
116     loop_control:
117       label: "{{ item.1 }}"
118     register: result
119     until: result is succeeded
120     retries: "{{ install_retries }}"
121     delay: "{{ install_delay }}"
122     when: item.0.module|default('package') == 'package'
123   - name: "packages: Debug Linux package"
124     ansible.builtin.debug:
125       var: result
126       when: cl_debug|bool
127   when: ansible_os_family == "RedHat" or
128         ansible_os_family == "Debian"
129 # Linux yum - - - - -
130 - block:
131   - name: "packages: Manage Linux packages by yum module"
132     ansible.builtin.yum:
133       name: "{{ item.name }}" # list / elements=string
134       state: "{{ item.state|default(omit) }}"
135       use_backend: "{{ item.use|default(omit) }}"
136       loop: "{{ cl_packages.values()|list }}"
137     loop_control:
138       label: "{{ item.name }}"
139     register: result
140     until: result is succeeded
141     retries: "{{ install_retries }}"
142     delay: "{{ install_delay }}"
143     when: item.module|default('package') == 'yum'
144   - name: "packages: Debug Linux yum"
145     ansible.builtin.debug:
146       var: result
147       when: cl_debug|bool
148   when: ansible_os_family == "RedHat" or
149         ansible_os_family == "Debian"
150

```

(continues on next page)

(continued from previous page)

```

151 # Linux apt - - - - -
152 - block:
153   - name: "packages: Manage Linux packages by apt module"
154     ansible.builtin.apt:
155       name: "{{ item.name }}" # list / elements=string
156       state: "{{ item.state|default(omit) }}"
157       purge: "{{ item.purge|default(omit) }}"
158       loop: "{{ cl_packages.values()|list }}"
159       loop_control:
160         label: "{{ item.name }}"
161       register: result
162       until: result is succeeded
163       retries: "{{ install_retries }}"
164       delay: "{{ install_delay }}"
165       when: item.module|default('package') == 'apt'
166   - name: "packages: Debug Linux apt"
167     ansible.builtin.debug:
168       var: result
169       when: cl_debug|bool
170   when: ansible_os_family == "RedHat" or
171         ansible_os_family == "Debian"
172 # TODO: Complete parameters of modules
173 # EOF
174 ...

```

4.1.12 states.yml

Synopsis: Configure states.

Description of the task.

[tasks/states.yml]

```

1 ---
2
3 - name: "states: Debug"
4   ansible.builtin.debug:
5     msg: "{{ cl_states }}"
6     when: cl_debug|bool
7
8 - name: "states: Apply unmount states"
9   ansible.posix.mount:
10    path: "{{ item.value.path }}"
11    state: "{{ item.value.state|default('absent') }}"
12    backup: "{{ item.value.backup|default(omit) }}"
13    boot: "{{ item.value.boot|default(omit) }}"
14    dump: "{{ item.value.dump|default(omit) }}"
15    fstab: "{{ item.value.fstab|default(omit) }}"
16    opts: "{{ item.value.opts|default(omit) }}"
17    passno: "{{ item.value.passno|default(omit) }}"
18    src: "{{ item.value.src|default(omit) }}"
19    loop: "{{ cl_states|dict2items }}"
20    loop_control:

```

(continues on next page)

(continued from previous page)

```

21     label: "{{ item.value.path }}"
22     when: item.value.state|default('absent') in cl_states_unmount
23
24 - name: "states: Apply file states"
25   ansible.builtin.file:
26     path: "{{ item.value.path }}"
27     state: "{{ item.value.state|default(omit) }}"
28     src: "{{ item.value.src|default(omit) }}"
29     owner: "{{ item.value.owner|default(omit) }}"
30     group: "{{ item.value.group|default(omit) }}"
31     mode: "{{ item.value.mode|default(omit) }}"
32     attributes: "{{ item.value.attributes|default(omit) }}"
33     recurse: "{{ item.value.recurse|default(omit) }}"
34     force: "{{ item.value.force|default(omit) }}"
35     follow: "{{ item.value.follow|default(omit) }}"
36     access_time: "{{ item.value.access_time|default(omit) }}"
37     access_time_format: "{{ item.value.access_time_format|default(omit) }}"
38     modification_time: "{{ item.value.modification_time|default(omit) }}"
39     modification_time_format: "{{ item.value.modification_time_format|default(omit) }}"
40     ↩↪ unsafe_writes: "{{ item.value.unsafe_writes|default(omit) }}"
41     loop: "{{ cl_states|dict2items }}"
42     loop_control:
43       label: "{{ item.value.path }}"
44     when: item.value.state|default('file') in cl_states_file
45
46 - name: "states: Apply mount states"
47   ansible.posix.mount:
48     path: "{{ item.value.path }}"
49     state: "{{ item.value.state }}"
50     backup: "{{ item.value.backup|default(omit) }}"
51     boot: "{{ item.value.boot|default(omit) }}"
52     dump: "{{ item.value.dump|default(omit) }}"
53     fstab: "{{ item.value.fstab|default(omit) }}"
54     opts: "{{ item.value.opts|default(omit) }}"
55     passno: "{{ item.value.passno|default(omit) }}"
56     src: "{{ item.value.src|default(omit) }}"
57     loop: "{{ cl_states|dict2items }}"
58     loop_control:
59       label: "{{ item.value.path }}"
60     when: item.value.state|default('absent') in cl_states_mount
61
62 # EOF
63 ...

```

4.1.13 files.yml

Synopsis: Manage files.

Iterate `cl_files_order` (6) and include tasks with particular modules.

[tasks/files.yml]

```

1 ---
2
3 - name: "files: Manage files"

```

(continues on next page)

(continued from previous page)

```

4  ansible.builtin.include_tasks:
5      file: "files-{{ fitem }}.yaml"
6  loop: "{{ cl_files_order }}"
7  loop_control:
8      loop_var: fitem
9
10 # EOF
11 ...

```

See also:

- See *Files Order of options*

Hint:

- Customize the list `cl_files_order` and fit the order of the options to your needs.

4.1.14 files-blockinfile.yml

Synopsis: Configure files-blockinfile.

Description of the task.

[tasks/files-blockinfile.yml]

```

1  ---
2
3  - name: "files-blockinfile: Debug blocks in cl_files.values()"
4    ansible.builtin.debug:
5      msg: "{{ cl_files.values()|selectattr('blocks', 'defined')|list }}"
6      when: cl_debug|bool
7
8  - name: "files-blockinfile: Blockinfile"
9    ansible.builtin.blockinfile:
10     path: "{{ item.0.path }}"
11     marker: "# {mark} ANSIBLE MANAGED BLOCK {{ item.1.marker }}"
12     block: "{{ item.1.block }}"
13     state: "{{ item.1.state|default(omit) }}"
14     insertafter: "{{ item.1.insertafter|default(omit) }}"
15     insertbefore: "{{ item.1.insertbefore|default(omit) }}"
16     marker_begin: "{{ item.1.marker_begin|default(omit) }}"
17     marker_end: "{{ item.1.marker_end|default(omit) }}"
18     owner: "{{ item.0.owner|default(omit) }}"
19     group: "{{ item.0.group|default(omit) }}"
20     mode: "{{ item.0.mode|default(omit) }}"
21     attributes: "{{ item.0.attributes|default(omit) }}"
22     create: "{{ item.0.create|default(omit) }}"
23     validate: "{{ item.0.validate|default(omit) }}"
24     # backup: "{{ cl_backup }}"
25     loop: "{{ cl_files.values()|list|subelements('blocks', skip_missing=true) }}"
26     loop_control:
27         label: "{{ item.0.path }} {{ item.1.marker }}"
28     notify: "{{ item.0.handlers|default(omit) }}"
29     register: cl_results_blocks
30

```

(continues on next page)

(continued from previous page)

```

31 - block:
32   - name: "files-blockinfile: Debug cl_results_blocks"
33     ansible.builtin.debug:
34       var: cl_results_blocks
35   - name: "files-blockinfile: Debug changed blocks paths"
36     ansible.builtin.debug:
37       msg: "{{ cl_results_blocks.results|default([])|
38             json_query('[?changed].invocation.module_args.path') }}"
39     when: cl_debug|bool
40
41 # EOF
42 ...

```

4.1.15 files-copy.yml

Synopsis: Configure files-copy.

Description of the task.

[tasks/files-copy.yml]

```

1 ---
2
3 - name: "files-copy: Debug copyfile in cl_files.values()"
4   ansible.builtin.debug:
5     msg: "{{ cl_files.values()|selectattr('copyfile', 'defined')|list }}"
6   when: cl_debug|bool
7
8 - name: "files-copy: Delete copyfile in cl_files.values()"
9   ansible.builtin.file:
10    state: absent
11    path: "{{ item.path }}"
12    loop: "{{ cl_files.values()|selectattr('copyfile', 'defined')|list }}"
13    loop_control:
14      label: "{{ item.path }}"
15    when: cl_copyfile_delete|bool
16
17 - name: "files-copy: Copy copyfile in cl_files.values()"
18   ansible.builtin.copy:
19     dest: "{{ item.path }}"
20     src: "{{ item.copyfile.path }}"
21     checksum: "{{ item.copyfile.checksum|default(omit) }}"
22     content: "{{ item.copyfile.content|default(omit) }}"
23     decrypt: "{{ item.copyfile.decrypt|default(omit) }}"
24     directory_mode: "{{ item.copyfile.directory_mode|default(omit) }}"
25     follow: "{{ item.copyfile.follow|default(omit) }}"
26     force: "{{ item.copyfile.force|default(omit) }}"
27     local_follow: "{{ item.copyfile.local_follow|default(omit) }}"
28     remote_src: "{{ item.copyfile.remote_src|default(omit) }}"
29     owner: "{{ item.owner|default(omit) }}"
30     group: "{{ item.group|default(omit) }}"
31     mode: "{{ item.mode|default(omit) }}"
32     attributes: "{{ item.attributes|default(omit) }}"
33     validate: "{{ item.validate|default(omit) }}"
34     # backup: "{{ cl_backup }}"
35     loop: "{{ cl_files.values()|selectattr('copyfile', 'defined')|list }}"

```

(continues on next page)

(continued from previous page)

```

36  loop_control:
37      label: "{{ item.path }}"
38  notify: "{{ item.handlers|default(omit) }}"
39  register: cl_results_copy
40
41  - block:
42      - name: "files-copy: Debug cl_results_copy"
43        ansible.builtin.debug:
44            var: cl_results_copy
45      - name: "files-copy: Debug changed copy path"
46        ansible.builtin.debug:
47            msg: "{{ cl_results_copy|default([])|
48                  json_query('[?changed].invocation.module_args.path') }}"
49        when: cl_debug|bool
50
51  # EOF
52  ...

```

4.1.16 files-create-backup.yml

Synopsis: Configure files-create-backup.

Description of the task.

[tasks/files-create-backup.yml]

```

1  ---
2
3  - name: "file-create-backup: Create time-stamp"
4    ansible.builtin.set_fact:
5        cl_timestamp: "{{ '%Y-%m-%d_%H_%M_%S'|strftime }}"
6
7  - name: "file-create-backup: Stat cl_files"
8    ansible.builtin.stat:
9        path: "{{ item }}"
10   loop: "{{ cl_files.values()|map(attribute='path')|list|unique }}"
11   loop_control:
12       label: "{{ item }}"
13   register: result
14
15  - name: "file-create-backup: Debug result"
16    ansible.builtin.debug:
17        var: result
18   when: cl_debug|bool
19
20  - name: "file-create-backup: Create backup files"
21    ansible.builtin.copy:
22        remote_src: true
23        src: "{{ item.item }}"
24        dest: "{{ item.item }}_{{ cl_timestamp }}.bak"
25        mode: preserve
26   loop: "{{ result.results }}"
27   loop_control:
28       label: "{{ item.item }}"
29   when: item.stat.exists
30   changed_when: false

```

(continues on next page)

(continued from previous page)

```

31
32 # EOF
33 ...

```

4.1.17 files-delete-backup.yml

Synopsis: Configure files-delete-backup.

Description of the task.

[tasks/files-delete-backup.yml]

```

1 ---
2
3 - name: "files-delete-backup: Delete backup files that did not change"
4   ansible.builtin.file:
5     state: absent
6     path: "{{ item }}_{{ cl_timestamp }}.bak"
7     loop: "{{ cl_files.values() | list | json_query('[] .path') |
8           difference(cl_results_copy.results | default([]) |
9                     json_query('[?changed].invocation.module_args.path')) | unique |
10          difference(cl_results_template.results | default([]) |
11                    json_query('[?changed].invocation.module_args.path')) | unique |
12          difference(cl_results_lines.results | default([]) |
13                    json_query('[?changed].invocation.module_args.path')) | unique |
14          difference(cl_results_blocks.results | default([]) |
15                    json_query('[?changed].invocation.module_args.path')) | unique |
16          difference(cl_results_inifile.results | default([]) |
17                    json_query('[?changed].invocation.module_args.path')) | unique |
18          difference(cl_results_ucl.results | default([]) |
19                    json_query('[?changed].invocation.module_args.path')) | unique |
20          difference(cl_results_patch.results | default([]) |
21                    json_query('[?changed].invocation.module_args.path')) | unique
22          }}"
23   when:
24     - cl_backup | bool
25     - not ansible_check_mode
26   changed_when: false
27
28 # EOF
29 ...

```

4.1.18 files-inifile.yml

Synopsis: Configure files-inifile.

Description of the task.

[tasks/files-inifile.yml]

```

1 ---
2
3 - name: "files-inifile: Debug ini in cl_files.values()"
4   ansible.builtin.debug:
5     msg: "{{ cl_files.values() | selectattr('ini', 'defined') | list }}"

```

(continues on next page)

(continued from previous page)

```

6   when: cl_debug|bool
7
8   - name: "files-inifile: INI files"
9     community.general.ini_file:
10      path: "{{ item.0.path }}"
11      section: "{{ item.1.section }}"
12      option: "{{ item.1.option|default(omit) }}"
13      value: "{{ item.1.value|default(omit) }}"
14      state: "{{ item.1.state|default(omit) }}"
15      allow_no_value: "{{ item.1.allow_no_value|default(omit) }}"
16      no_extra_spaces: "{{ item.1.no_extra_spaces|default(omit) }}"
17      owner: "{{ item.0.owner|default(omit) }}"
18      group: "{{ item.0.group|default(omit) }}"
19      mode: "{{ item.0.mode|default(omit) }}"
20      attributes: "{{ item.0.attributes|default(omit) }}"
21      create: "{{ item.0.create|default(omit) }}"
22      # backup: "{{ cl_backup }}"
23      loop: "{{ cl_files.values()|list|subelements('ini', {'skip_missing': True}) }}"
24      loop_control:
25        label: "{{ item.0.path }}"
26      notify: "{{ item.0.handlers|default(omit) }}"
27      register: cl_results_ini
28
29   - block:
30     - name: "files-inifile: Debug cl_results_ini"
31       ansible.builtin.debug:
32         var: cl_results_ini
33     - name: "files-inifile: Debug changed ini paths"
34       ansible.builtin.debug:
35         msg: "{{ cl_results_ini.results|default([])|
36              json_query('[?changed].invocation.module_args.path') }}"
37       when: cl_debug|bool
38
39   # EOF
40   ...

```

4.1.19 files-lineinfile.yml

Synopsis: Configure files-lineinfile.

Description of the task.

[tasks/files-lineinfile.yml]

```

1   ---
2
3   # lines
4   - name: "files-lineinfile: Debug lines in cl_files.values()"
5     ansible.builtin.debug:
6       msg: "{{ cl_files.values()|selectattr('lines', 'defined')|list }}"
7     when: cl_debug|bool
8
9   - name: "files-lineinfile: Lineinfile lines"
10    ansible.builtin.lineinfile:
11      path: "{{ item.0.path }}"
12      regexp: "{{ item.1.regexp|default(omit) }}"

```

(continues on next page)

(continued from previous page)

```

13   line: "{{ item.1.line|default(omit) }}"
14   backrefs: "{{ item.1.backrefs|default(omit) }}"
15   state: "{{ item.1.state|default(omit) }}"
16   firstmatch: "{{ item.1.firstmatch|default(omit) }}"
17   insertafter: "{{ item.1.insertafter|default(omit) }}"
18   insertbefore: "{{ item.1.insertbefore|default(omit) }}"
19   owner: "{{ item.0.owner|default(omit) }}"
20   group: "{{ item.0.group|default(omit) }}"
21   mode: "{{ item.0.mode|default(omit) }}"
22   attributes: "{{ item.0.attributes|default(omit) }}"
23   others: "{{ item.0.others|default(omit) }}"
24   create: "{{ item.0.create|default(omit) }}"
25   validate: "{{ item.0.validate|default(omit) }}"
26   # backup: "{{ cl_backup }}"
27   loop: "{{ cl_files.values()|list|subelements('lines', skip_missing=true) }}"
28   loop_control:
29     label: "{{ item.0.path }} {{ item.1.line|default('') }}"
30   notify: "{{ item.0.handlers|default(omit) }}"
31   register: cl_results_lines
32
33 - block:
34   - name: "files-lineinfile: Debug cl_results_lines"
35     ansible.builtin.debug:
36       var: cl_results_lines
37   - name: "files-lineinfile: Debug changed lines paths"
38     ansible.builtin.debug:
39       msg: "{{ cl_results_lines.results|default([])|
40             json_query('[?changed].invocation.module_args.path') }}"
41   when: cl_debug|bool
42
43 # dict
44 - name: "files-lineinfile: Debug dict in cl_files.values()"
45   ansible.builtin.debug:
46     msg: "{{ cl_files.values()|selectattr('dict', 'defined')|list }}"
47   when: cl_debug|bool
48
49 - name: "files-lineinfile: Lineinfile dict"
50   ansible.builtin.lineinfile:
51     path: "{{ item.0.path }}"
52     regexp: '^\\s*[#;]*\\s*{{ item.1.key }}\\s*{{ item.0.assignment|default("=")|trim }}
↪ \\s*(.*)$'
53     line: "{{ item.1.key }}{{ item.0.assignment|default('=') }}{{ item.1.value }}"
54     backrefs: "{{ item.1.backrefs|default(omit) }}"
55     state: "{{ item.1.state|default(omit) }}"
56     firstmatch: "{{ item.1.firstmatch|default(omit) }}"
57     insertafter: "{{ item.1.insertafter|default(omit) }}"
58     insertbefore: "{{ item.1.insertbefore|default(omit) }}"
59     owner: "{{ item.0.owner|default(omit) }}"
60     group: "{{ item.0.group|default(omit) }}"
61     mode: "{{ item.0.mode|default(omit) }}"
62     attributes: "{{ item.0.attributes|default(omit) }}"
63     others: "{{ item.0.others|default(omit) }}"
64     create: "{{ item.0.create|default(omit) }}"
65     validate: "{{ item.0.validate|default(omit) }}"
66     # backup: "{{ cl_backup }}"
67     loop: "{{ cl_files.values()|list|subelements('dict', skip_missing=true) }}"
68     loop_control:

```

(continues on next page)

(continued from previous page)

```

69     label: "{{ item.0.path }} {{ item.1.key }}"
70     notify: "{{ item.0.handlers|default(omit) }}"
71     register: cl_results_dict
72
73 - block:
74   - name: "files-lineinfile: Debug cl_results_dict"
75     ansible.builtin.debug:
76       var: cl_results_dict
77   - name: "files-lineinfile: Debug changed dict paths"
78     ansible.builtin.debug:
79       msg: "{{ cl_results_dict.results|default([])|
80              json_query('[?changed].invocation.module_args.path') }}"
81     when: cl_debug|bool
82
83 # EOF
84 ...

```

4.1.20 files-markers.yml

Synopsis: Configure files-markers.

Description of the task.

[tasks/files-markers.yml]

```

1 ---
2
3 - name: "files-markers: Debug markers in cl_files.values()"
4   ansible.builtin.debug:
5     msg: "{{ cl_files.values()|selectattr('markers', 'defined')|list }}"
6   when: cl_debug|bool
7
8 - name: "files-markers: Mark block {{ item.1.marker }}"
9   ansible.builtin.include_tasks: fn/mark-block.yml
10  loop: "{{ cl_files.values()|list|subelements('markers', skip_missing=true) }}"
11  loop_control:
12    label: "{{ item.0.path }}"
13
14 # EOF
15 ...

```

4.1.21 mark-block.yml

Synopsis: Configure mark-block.

Description of the task.

[tasks/fn/mark-block.yml]

```

1 ---
2
3 - name: "mark-block: Check begin marker {{ item.1.marker }}"
4   ansible.builtin.command:
5     cmd: >
6     grep -q '# BEGIN ANSIBLE MANAGED BLOCK {{ item.1.marker }}' {{ item.0.path }}

```

(continues on next page)

(continued from previous page)

```

7  register: checkmarker
8  ignore_errors: true
9  changed_when: false
10
11 - block:
12   - name: "mark-block: Create begin marker {{ item.1.marker }}"
13     ansible.builtin.replace:
14       path: '{{ item.0.path }}'
15       regexp: '{{ item.1.regex1 }}'
16       replace: |-
17         {{ '#' }} BEGIN ANSIBLE MANAGED BLOCK {{ item.1.marker }}
18         {{ item.1.replace1 }}
19   - name: "mark-block: Create end marker {{ item.1.marker }}"
20     ansible.builtin.replace:
21       path: '{{ item.0.path }}'
22       regexp: '({{ item.1.regex1 }}[\s\S]*?){{ item.1.regex2 }}'
23       replace: |-
24         \g<1>
25         {{ item.1.replace2 }}
26         {{ '#' }} END ANSIBLE MANAGED BLOCK {{ item.1.marker }}
27
28   when:
29     - not ansible_check_mode
30     - checkmarker.rc != 0
31
32 # EOF
...

```

4.1.22 files-patch.yml

Synopsis: Configure files-patch.

Description of the task.

[tasks/files-patch.yml]

```

1  ---
2
3  - name: "files-patch: Debug patch in cl_files.values()"
4    ansible.builtin.debug:
5      msg: "{{ cl_files.values()|selectattr('patch', 'defined')|list }}"
6      when: cl_debug|bool
7
8  - name: "files-patch: Patch"
9    ansible.posix.patch:
10     dest: "{{ item.path }}"
11     src: "{{ item.patch.src }}"
12     basedir: "{{ item.patch.basedir|default(omit) }}"
13     binary: "{{ item.patch.binary|default(omit) }}"
14     ignore_whitespace: "{{ item.patch.ignore_whitespace|default(omit) }}"
15     remote_src: "{{ item.patch.remote_src|default(omit) }}"
16     state: "{{ item.patch.state|default(omit) }}"
17     strip: "{{ item.patch.strip|default(omit) }}"
18     # backup: "{{ cl_backup }}"
19     loop: "{{ cl_files.values()|selectattr('patch', 'defined')|list }}"
20     loop_control:
21       label: "{{ item.path }}"

```

(continues on next page)

(continued from previous page)

```

22  notify: "{{ item.handlers|default(omit) }}"
23  register: cl_results_patch
24
25 - block:
26   - name: "files-patch: Debug cl_results_patch"
27     ansible.builtin.debug:
28       var: cl_results_patch
29   - name: "files-patch: Debug changed patch patch"
30     ansible.builtin.debug:
31       msg: "{{ cl_results_patch|default([])|
32            json_query('[?changed].invocation.module_args.path') }}"
33   when: cl_debug|bool
34
35 # EOF
36 ...

```

4.1.23 files-template.yml

Synopsis: Configure files-template.

Description of the task.

[tasks/files-template.yml]

```

1  ---
2
3  - name: "files-template: Debug template in cl_files.values()"
4    ansible.builtin.debug:
5      msg: "{{ cl_files.values()|selectattr('template', 'defined')|list }}"
6    when: cl_debug|bool
7
8  - name: "files-template: Delete template in cl_files.values()"
9    ansible.builtin.file:
10     state: absent
11     path: "{{ item.path }}"
12     loop: "{{ cl_files.values()|selectattr('template', 'defined')|list }}"
13     loop_control:
14       label: "{{ item.path }}"
15     when: cl_template_delete|bool
16
17 - name: "files-template: Template"
18   ansible.builtin.template:
19     dest: "{{ item.path }}"
20     src: "{{ item.template.path }}"
21     follow: "{{ item.template.follow|default(omit) }}"
22     force: "{{ item.template.force|default(omit) }}"
23     block_start_string: "{{ item.template.block_start_string|default(omit) }}"
24     block_end_string: "{{ item.template.block_end_string|default(omit) }}"
25     lstrip_blocks: "{{ item.template.lstrip_blocks|default(omit) }}"
26     newline_sequence: "{{ item.template.newline_sequence|default(omit) }}"
27     output_encoding: "{{ item.template.output_encoding|default(omit) }}"
28     trim_blocks: "{{ item.template.trim_blocks|default(omit) }}"
29     variable_end_string: "{{ item.template.variable_end_string|default(omit) }}"
30     variable_start_string: "{{ item.template.variable_start_string|default(omit) }}"
31     owner: "{{ item.owner|default(omit) }}"
32     group: "{{ item.group|default(omit) }}"

```

(continues on next page)

(continued from previous page)

```

33     mode: "{{ item.mode|default(omit) }}"
34     attributes: "{{ item.attributes|default(omit) }}"
35     validate: "{{ item.validate|default(omit) }}"
36     # backup: "{{ cl_backup }}"
37     loop: "{{ cl_files.values()|selectattr('template', 'defined')|list }}"
38     loop_control:
39         label: "{{ item.path }}"
40     notify: "{{ item.handlers|default(omit) }}"
41     register: cl_results_template
42
43 - block:
44     - name: "files-template: Debug cl_results_template"
45       ansible.builtin.debug:
46         var: cl_results_template
47     - name: "files-template: Debug changed template path"
48       ansible.builtin.debug:
49         msg: "{{ cl_results_template|default([])|
50             json_query('[?changed].invocation.module_args.path') }}"
51     when: cl_debug|bool
52
53 # EOF
54 ...

```

4.1.24 files-ucl.yml

Synopsis: Configure files-ucl.

Description of the task.

[tasks/files-ucl.yml]

```

1 ---
2
3 - name: "files-ucl: Debug ucl in cl_files.values()"
4   ansible.builtin.debug:
5     msg: "{{ cl_files.values()|selectattr('ucl', 'defined')|list }}"
6   when: cl_debug|bool
7
8 - name: "files-ucl: UCL files cl_files.values()"
9   vbotka.freebsd.ucl:
10    path: "{{ item.0.path }}"
11    chdir: "{{ item.1.chdir|default(omit) }}"
12    upath: "{{ item.1.upath|default(omit) }}"
13    ipath: "{{ item.1.ipath|default(omit) }}"
14    icontent: "{{ item.1.icontent|default(omit) }}"
15    value: "{{ item.1.value|default(omit) }}"
16    vtype: "{{ item.1.vtype|default(omit) }}"
17    merge: "{{ item.1.merge|default(omit) }}"
18    state: "{{ item.1.state|default(omit) }}"
19    delimiter: "{{ item.1.delimiter|default(omit) }}"
20    lang: "{{ item.1.lang|default(omit) }}"
21    owner: "{{ item.0.owner|default(omit) }}"
22    group: "{{ item.0.group|default(omit) }}"
23    mode: "{{ item.0.mode|default(omit) }}"
24    attributes: "{{ item.0.attributes|default(omit) }}"
25    create: "{{ item.0.create|default(omit) }}"

```

(continues on next page)

(continued from previous page)

```

26     validate: "{{ item.0.validate|default(omit) }}"
27     # backup: "{{ cl_backup }}"
28     loop: "{{ cl_files.values()|list|subelements('ucl', {'skip_missing': True}) }}"
29     loop_control:
30         label: "{{ item.0.path }}"
31     notify: "{{ item.0.handlers|default(omit) }}"
32     register: cl_results_ucl
33
34 - block:
35     - name: "files-ucl: Debug cl_results_ucl"
36       ansible.builtin.debug:
37         var: cl_results_ucl
38     - name: "files-ucl: Debug changed ucl paths"
39       ansible.builtin.debug:
40         msg: "{{ cl_results_ucl.results|default([])|
41              json_query('[?changed].invocation.module_args.path') }}"
42     when: cl_debug|bool
43
44 # EOF
45 ...

```

4.1.25 services.yml

Synopsis: Configure services.

Description of the task.

[tasks/services.yml]

```

1  ---
2
3  - name: "services: Debug"
4    vars:
5      msg: |-
6          cl_services_freebsd_rcconf_auto [{{ cl_services_freebsd_rcconf_auto }}]
7          cl_services
8          {{ cl_services|to_nice_yaml }}
9    ansible.builtin.debug:
10     msg: "{{ '{ }'.format(msg) }}"
11     when: cl_debug|bool
12
13 # FreeBSD auto -----
14 - block:
15
16     - name: "services: Enable service in rc.conf FreeBSD"
17       ansible.builtin.lineinfile:
18         dest: /etc/rc.conf
19         regexp: '^s*{{ item.value.name }}_enable\s*=(.*)$'
20         line: '{{ item.value.name }}_enable="YES"'
21         backup: "{{ cl_backup }}"
22         loop: "{{ cl_services|dict2items }}"
23         loop_control:
24             label: "{{ item.key }}"
25         when: item.value.enabled|default(true)|bool
26
27     - name: "services: Disable service in rc.conf FreeBSD"

```

(continues on next page)

(continued from previous page)

```

28  ansible.builtin.lineinfile:
29      dest: /etc/rc.conf
30      regexp: '^^\s*{{ item.value.name }}_enable\s*=(.*)$'
31      line: '{{ item.value.name }}_enable="NO"'
32      backup: "{{ cl_backup }}"
33      loop: "{{ cl_services|dict2items }}"
34      loop_control:
35          label: "{{ item.key }}"
36      when: not item.value.enabled|default(true)|bool
37
38  when:
39      - ansible_os_family == 'FreeBSD'
40      - cl_services_freebsd_rcconf_auto|bool
41
42  # All -----
43  - block:
44
45      - name: "services: Manage services"
46        ansible.builtin.service:
47            name: "{{ item.value.name }}"
48            state: "{{ item.value.state|default('started') }}"
49            enabled: "{{ item.value.enabled|default(true) }}"
50            loop: "{{ cl_services|dict2items }}"
51            loop_control:
52                label: "{{ item.key }}"
53
54        when: ansible_os_family == 'RedHat' or
55              ansible_os_family == 'Debian' or
56              ansible_os_family == 'FreeBSD'
57
58  # EOF
59  ...

```

4.2 Handlers

4.2.1 main.yml

Synopsis: Main task.

Import handlers. Content of this file is created dynamically.

[handlers/main.yml]

See also:

- See *Best practice*

4.3 Templates

4.3.1 1.0.0-wpa_action.sh.j2

Synopsis: Template 1.0.0-wpa_action.sh.

Description of the template.

[templates/1.0.0-wpa_action.sh.j2]

```

1  #!/bin/sh
2  # {{ ansible_managed }}
3
4  version="1.0.0"
5  ifname=$1
6  cmd=$2
7  # TODO: test prams
8
9  {% if wpacli_action_script_log_to_file %}
10 logtofile="1"
11 {% else %}
12 logtofile="0"
13 {% endif %}
14 logfile="{{ wpacli_action_script_logfile }}"
15
16 if [ "$logtofile" = "1" ]; then
17     my_date=`date +%b %d %T`
18     printf "$my_date $ifname: $cmd \n" >> $logfile
19 fi
20
21 # wpa_supplicant reports connection to SSID. Start dhclient and
22 # restart routing
23 if [ "$cmd" = "CONNECTED" ]; then
24     if [ "$logtofile" = "1" ]; then
25         ssid=`wpa_cli -i$ifname status | grep ^ssid= | cut -f2- -d=`
26         printf "$my_date $ifname: SSID: $ssid \n" >> $logfile
27     fi
28     message=`/etc/rc.d/dhclient forrestart $ifname 2>&1` # NOTE 1
29     if [ "$logtofile" = "1" ]; then
30         printf "$my_date $ifname: dhclient start: $message \n" >> $logfile
31     fi
32     message=`/etc/rc.d/routing restart 2>&1`
33     if [ "$logtofile" = "1" ]; then
34         printf "$my_date $ifname: routing restart: $message \n" >> $logfile
35     fi
36 fi
37
38 # wpa_supplicant reports disconnection from SSID. Stop dhclient and
39 # restart routing
40 if [ "$cmd" = "DISCONNECTED" ]; then
41     message=`/etc/rc.d/dhclient forrestart $ifname`
42     if [ "$logtofile" = "1" ]; then
43         printf "$my_date $ifname: dhclient forrestart: $message \n" >> $logfile
44     fi
45     message=`/etc/rc.d/routing restart 2>&1`
46     if [ "$logtofile" = "1" ]; then
47         printf "$my_date $ifname: routing restart: $message \n" >> $logfile
48     fi
49 fi
50
51 exit 0
52
53 # NOTE 1
54 # We don't want /etc/network.subr to handle DHCP and instruct

```

(continues on next page)

(continued from previous page)

```

55 # ifconfig, in rc.conf, to configure WPA only.
56 #     ifconfig_wlan1="WPA"
57 # When we use "/etc/rc.d/dhclient start wlan1" it complains
58 #     'wlan1' is not a DHCP-enabled interface
59 # Hence we use /etc/rc.d/dhclient forrestart $ifname in this script
60
61 # NOTE 2
62 # Example how to activate the script.
63 # wpa_cli -B -i wlan1 -a /root/bin/wpa_action.sh
64
65 # EOF

```

4.3.2 1.1.0-wpa_action.sh.j2

Synopsis: Template 1.1.0-wpa_action.sh.

Description of the template.

[templates/1.1.0-wpa_action.sh.j2]

```

1  #!/bin/sh
2  # {{ ansible_managed }}
3
4  version="1.1.0"
5  ifname=$1
6  cmd=$2
7  # TODO: test prams
8
9  {% if wpacli_action_script_log_to_file %}
10 logtofile="1"
11 {% else %}
12 logtofile="0"
13 {% endif %}
14 logfile="{{ wpacli_action_script_logfile }}"
15
16 {% if wpacli_action_script_ntp_set %}
17 ntp_set="1"
18 {% else %}
19 ntp_set="0"
20 {% endif %}
21 ntp_server="{{ wpacli_action_script_ntp_server }}"
22 ntpdate_flags="{{ wpacli_action_script_ntpdate_flags }}"
23
24 # functions
25 log() {
26     if [ "$logtofile" = "1" ]; then
27         my_date=`date +%b %d %T`
28         printf "$my_date $ifname: $cmd: $message \n" >> $logfile
29     fi
30 }
31
32 routing_restart() {
33     cmd="/etc/rc.d/routing restart"
34     message=`$cmd 2>&1`
35     log
36 }

```

(continues on next page)

(continued from previous page)

```

37
38 dhclient_forcestart() {
39     cmd="/etc/rc.d/dhclient forcestart $ifname" # NOTE 1
40     message=`$cmd 2>&1`
41     log
42 }
43
44 dhclient_forcestop() {
45     cmd="/etc/rc.d/dhclient forcestop $ifname"
46     message=`$cmd 2>&1`
47     log
48 }
49
50 ntpd_stop() {
51     cmd="/etc/rc.d/ntpd stop"
52     message=`$cmd 2>&1`
53     log
54 }
55
56 ntpd_start() {
57     cmd="/etc/rc.d/ntpd start"
58     message=`$cmd 2>&1`
59     log
60 }
61
62 ntpdate_settimeofday() {
63     cmd="/usr/sbin/ntpdate $ntpdate_flags $ntp_server"
64     message=`$cmd 2>&1`
65     log
66 }
67
68 log_SSID() {
69     if [ "$logtofile" = "1" ]; then
70         ssid=`wpa_cli -i$ifname status | grep ^ssid= | cut -f2- -d=`
71         my_date=`date +"%b %d %T"`
72         printf "$my_date $ifname: SSID: $ssid \n" >> $logfile
73     fi
74 }
75
76 # log interface and command
77 if [ "$logtofile" = "1" ]; then
78     my_date=`date +"%b %d %T"`
79     printf "$my_date $ifname: $cmd \n" >> $logfile
80 fi
81
82 # wpa_supplicant reports connection to SSID. Start dhclient and
83 # restart routing
84 if [ "$cmd" = "CONNECTED" ]; then
85     log_SSID
86     dhclient_forcestart
87     routing_restart
88     if [ "$ntp_set" = "1" ]; then # NOTE 3
89         ntpd_stop
90         ntpdate_settimeofday
91         ntpd_start
92     fi
93 fi

```

(continues on next page)

(continued from previous page)

```

94
95 # wpa_supplicant reports disconnection from SSID. Stop dhclient and
96 # restart routing
97 if [ "$cmd" = "DISCONNECTED" ]; then
98     dhclient_forcestop
99     routing_restart
100 fi
101
102 exit 0
103
104 # NOTE 1
105 # We don't want /etc/network.subr to handle DHCP. Therefore we instruct
106 # ifconfig, in rc.conf, to configure WPA only
107 #     ifconfig_wlan1="WPA"
108 # When we use "/etc/rc.d/dhclient start wlan1" it complains
109 #     'wlan1' is not a DHCP-enabled interface
110 # Hence we use /etc/rc.d/dhclient forcestart $ifname in this script.
111
112 # NOTE 2
113 # Example how to activate the script
114 # wpa_cli -B -i wlan1 -a /root/bin/wpa_action.sh
115
116 # NOTE 3
117 # In a wifi-only system, /etc/rc.d/ntpdate will time-out if it
118 # executes before /etc/rc.d/wpa_supplicant connects to the network
119 # (See rcorder /etc/rc.d/*)
120
121 # EOF

```

4.3.3 dma-auth.conf.j2

Synopsis: Template dma-auth.conf.

Description of the template.

[templates/dma-auth.conf.j2]

```

1 # {{ ansible_managed }}
2 # Format: myuser|smtp.gmail.com:mypassword
3 {% for item in cl_dma_authconf %}
4 {{ item }}
5 {% endfor %}
6 # EOF

```

4.3.4 handlers-auto1.yml.j2

Synopsis: Template handlers-auto1.yml.

Description of the template.

[templates/handlers-auto1.yml.j2]

```

1 ---
2 # {{ ansible_managed }}
3 # Automatically generated file with handlers.

```

(continues on next page)

(continued from previous page)

```
4 {% for iitem in item.value.handlers %}
5
6 - name: {{ item.key }} {{ iitem.handler }}
7   {{ iitem.module }}:
8 {% for param in iitem.params %}
9   {{ param }}
10 {% endfor %}
11 {% if iitem.conditions is defined %}
12   when:
13 {% for condition in iitem.conditions %}
14   {{ condition }}
15 {% endfor %}
16 {% endif %}
17
18 {% endfor %}
19
20 # EOF
```

4.3.5 handlers-auto2.yml.j2

Synopsis: Template handlers-auto2.yml.

Description of the template.

[templates/handlers-auto2.yml.j2]

```
1 ---
2 # {{ ansible_managed }}
3 # Automatically generated file with handlers.
4 {% for iitem in item.value.handlers %}
5
6 - name: {{ iitem.handler }}
7   {{ iitem.module }}:
8 {% for param in iitem.params %}
9   {{ param }}
10 {% endfor %}
11 {% if iitem.conditions is defined %}
12   when:
13 {% for condition in iitem.conditions %}
14   {{ condition }}
15 {% endfor %}
16 {% endif %}
17
18 {% endfor %}
19
20 # EOF
```

4.3.6 hosts.j2

Synopsis: Template hosts.

Description of the template.

[templates/hosts.j2]

```

1 # {{ ansible_managed }}
2 {{ cl_hosts_localhost_IPv6 }}      localhost localhost.{{ cl_domain }}
3 {{ cl_hosts_localhost_IPv4 }}      localhost localhost.{{ cl_domain }}
4
5 {% for item in cl_hosts %}
6 {{ item.ip }}      {{ item.fqdn }} {{ item.hostname|default('') }}
7 {% endfor %}
8
9 # EOF

```

4.3.7 loader.conf.j2

Synopsis: Template loader.conf.

Description of the template.

[templates/loader.conf.j2]

```

1 # {{ ansible_managed }}
2
3 # bsd_cimage_loaderconf_data
4 {% for item in cl_loaderconf_data %}
5 {{ item }}
6 {% endfor %}
7
8 # bsd_cimage_loaderconf_sysctl
9 {% for item in cl_loaderconf_sysctl %}
10 {{ item }}
11 {% endfor %}
12
13 # bsd_cimage_loaderconf_modules
14 {% for item in cl_loaderconf_modules %}
15 {{ item }}_load="YES"
16 {% endfor %}
17
18 # EOF

```

4.3.8 mailer.conf.j2

Synopsis: Template mailer.conf.

Description of the template.

[templates/mailer.conf.j2]

```

1 # {{ ansible_managed }}
2 # https://www.freebsd.org/cgi/man.cgi?mailer.conf
3 {% for item in cl_mailerconf %}
4 {{ item }}
5 {% endfor %}
6 # EOF

```

4.3.9 ntp.conf-minimal.j2

Synopsis: Template ntp.conf-minimal.

Description of the template.

[templates/ntp.conf-minimal.j2]

```

1 # {{ ansible_managed }}
2 {% for option in fp_ntp_config_options %}
3 {{ option }}
4 {% endfor %}
5 {% for pool in fp_ntp_config_pool %}
6 pool {{ pool }}
7 {% endfor %}
8 {% for restrict in fp_ntp_config_restrict %}
9 restrict {{ restrict }}
10 {% endfor %}
11 {% if fp_ntp_config_leapfile %}
12 leapfile {{ fp_ntp_config_leapfile }}
13 {% endif %}

```

4.3.10 revaliases.j2

Synopsis: Template revaliases.

Description of the template.

[templates/revaliases.j2]

```

1 # {{ ansible_managed }}
2 # Format: local_account:outgoing_address:mailhub
3 {% for item in cl_ssmtp_revaliases %}
4 {{ item }}
5 {% endfor %}
6 # EOF

```

4.3.11 ssmtp.conf.j2

Synopsis: Template ssmtp.conf.

Description of the template.

[templates/ssmtp.conf.j2]

```

1 # {{ ansible_managed }}
2
3 # The user that gets all the mails (UID < 1000, usually the admin)
4 # root=username@gmail.com
5 root={{ cl_ssmtp_postmaster_address }}
6
7 # The mail server (where the mail is sent to), both port 465 or 587 should be
8 ↪ acceptable
9 # See also https://support.google.com/mail/answer/78799
10 # mailhub=smtp.gmail.com:587
11 mailhub={{ cl_ssmtp_mailhub }}
12
13 # The address where the mail appears to come from for user authentication.
14 # rewriteDomain=gmail.com
15 rewriteDomain={{ cl_ssmtp_rewriteDomain }}

```

(continues on next page)

(continued from previous page)

```

15
16 # The full hostname.  Must be correctly formed, fully qualified domain
17 # name or GMail will reject connection.
18 # hostname=yourlocalhost.yourlocaldomain.tld
19 hostname={{ cl_ssmtp_srv }}
20
21 # Use SSL/TLS before starting negotiation
22 #UseTLS=Yes
23 #UseSTARTTLS=Yes
24 UseTLS={{ cl_ssmtp_UseTLS }}
25 UseSTARTTLS={{ cl_ssmtp_UseSTARTTLS }}
26
27 # Username/Password
28 #AuthUser=username
29 #AuthPass=password
30 #AuthMethod=LOGIN
31 AuthUser={{ cl_ssmtp_AuthUser }}
32 AuthPass={{ cl_ssmtp_AuthPass }}
33 AuthMethod={{ cl_ssmtp_AuthMethod }}
34
35 # Email 'From header's can override the default domain?
36 #FromLineOverride=yes
37 FromLineOverride={{ cl_ssmtp_FromLineOverride }}
38
39 # EOF

```

4.3.12 wpa_cli.j2

Synopsis: Template wpa_cli.

Description of the template.

[templates/wpa_cli.j2]

```

1 #!/bin/sh
2 # {{ ansible_managed }}"
3
4 # PROVIDE: wpa_cli
5 # REQUIRE: mountcritremote
6 # KEYWORD: nojail nostart
7
8 . /etc/rc.subr
9 . /etc/network.subr
10
11 name="wpa_cli"
12 desc="Frontend to WPA/802.11i Supplicant for wireless network
13 devices. Run in daemon mode executing the action file based on events
14 from wpa_supplicant"
15 rcvar=
16
17 ifn="$2"
18 if [ -z "$ifn" ]; then
19     return 1
20 fi
21
22 load_rc_config $name

```

(continues on next page)

(continued from previous page)

```

23
24 command="${wpa_cli_program}"
25 pidfile="/var/run/${name}/${ifn}.pid"
26 command_args="-B -i $ifn -P $pidfile -p ${wpa_cli_ctrl_interface} -a ${wpa_cli_action_
  →file}"
27 required_files="${wpa_cli_action_file}"
28
29 run_rc_command "$1"
30
31 # EOF

```

4.3.13 wpa_supplicant.conf.j2

Synopsis: Template wpa_supplicant.conf.

Description of the template.

[templates/wpa_supplicant.conf.j2]

```

1 # {{ ansible_managed }}
2 {% for gvar in cl_wpasupconf_global %}
3   {{ gvar.key }}={{ gvar.value }}
4 {% endfor %}
5 {% for net in item.network %}
6   {% for nvar in net.conf %}
7     {% if loop.first %}
8
9     network={
10      {% endif %}
11        {{ nvar.key }}={{ nvar.value }}
12    {% if loop.last %}
13    }
14  {% endif %}
15  {% endfor %}
16  {% endfor %}
17
18 # EOF

```

4.3.14 wpa_supplicant.conf.wlan0.j2

Synopsis: Template wpa_supplicant.conf.wlan0.

Description of the template.

[templates/wpa_supplicant.conf.wlan0.j2]

```

1 # {{ ansible_managed }}
2 {% for gvar in cl_wpasupconf_global %}
3   {{ gvar.key }}={{ gvar.value }}
4 {% endfor %}
5 {% for net in cl_wpasupconf_wlan0.network %}
6   {% for nvar in net.conf %}
7     {% if loop.first %}
8
9     network={

```

(continues on next page)

(continued from previous page)

```
10 {% endif %}
11     {{ nvar.key }}={{ nvar.value }}
12 {% if loop.last %}
13 }
14 {% endif %}
15 {% endfor %}
16 {% endfor %}
17
18 # EOF
```


5.1 vbotka.freebsd

- **Galaxy:** vbotka.freebsd
- **GitHub:** vbotka.freebsd

5.1.1 Documented roles

- ansible
- ansible_runner
- apache
- config_light
- freebsd_custom_image
- freebsd_postinstall
- freebsd_poudriere
- freebsd_wpa_cli

CHAPTER 6

Copyright

Redistribution and use in source (RST DocUtils) and ‘compiled’ forms (XML, HTML, PDF, PostScript, RTF and so forth) with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code (RST Docutils) must retain the above copyright notice, this list of conditions and the following disclaimer as the first lines of this file unmodified.

Redistributions in compiled form (transformed to other DTDs, converted to PDF, PostScript, RTF and other formats) must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

Important: THIS DOCUMENTATION IS PROVIDED “AS IS” AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE PROVIDER BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS DOCUMENTATION, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

CHAPTER 7

Legal Notice

All product names, logos, and brands are property of their respective owners.

CHAPTER 8

Indices and tables

- `genindex`
- `modindex`
- `search`